

z/OS



Security Server RACF Command Language Reference

z/OS



Security Server RACF Command Language Reference

Note

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 519.

Fourth Edition, September 2002

This is a major revision of SA22–7687–02.

This edition applies to Version 1 Release 4 of z/OS (5694-A01), Version 1 Release 4 of z/OS.e (5655-G52) and to all subsequent releases and modifications until otherwise indicated in new editions.

Order documents through your IBM® representative or the IBM branch office serving your locality. Documents are not stocked at the address below.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this document, or you may address your comments to the following address:

International Business Machines Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States & Canada): 1+845+432-9405

FAX (Other Countries):

Your International Access Code +1+845+432-9405

IBMLink™ (United States customers only): IBMUSM10(MHVRCFS)

Internet e-mail: mhvrfs@us.ibm.com

World Wide Web: <http://www.ibm.com/servers/eserver/zseries/zos/webqs.html>

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this document
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1994, 2002. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	vii
Tables	ix
About this document	xi
Purpose of this document	xi
Who should use this document	xi
How to use this document	xi
Where to find more information	xii
Softcopy publications	xii
RACF courses	xii
Using LookAt to look up message explanations	xiii
Accessing z/OS licensed documents on the Internet	xiii
IBM systems center publications	xiv
Other sources of information	xiv
IBM discussion areas	xiv
Internet sources	xiv
To request copies of IBM publications	xvi
Summary of changes	xvii
Chapter 1. Introduction	1
Summary of commands and their functions	1
Chapter 2. Basic information for issuing RACF commands	7
How to enter RACF commands	7
RACF TSO commands	7
RACF operator commands	7
Command direction and automatic command direction	7
RACF parameter library	8
R_admin callable service	8
Summary	8
Syntax of RACF commands and operands	9
Return codes from RACF commands	11
RACF command restriction for non-base segments in RACF profiles	11
Installation exit routines from RACF commands	12
Attribute and authority summary	12
Group authorities	13
Access authority for data sets	13
Chapter 3. RACF TSO commands	15
How to enter RACF TSO commands	15
Choosing between Using RACF TSO Commands and ISPF Panels	16
Entering RACF TSO commands in the foreground	17
Entering RACF TSO commands in the background	18
Chapter 4. RACF operator commands	21
Rules for entering RACF operator commands	21
Chapter 5. RACF Command Syntax	23
ADDGROUP (Add Group Profile)	24
ADDSD (Add Data Set Profile)	33
ADDUSER (Add User Profile)	48

ALTDSD (Alter Data Set Profile)	90
ALTGROUP (Alter Group Profile)	104
ALTUSER (Alter User Profile)	115
CONNECT (Connect User to Group)	173
DELDSD (Delete Data Set Profile)	181
DELGROUP (Delete Group Profile)	186
DELUSER (Delete User Profile)	189
DISPLAY (Display Signed-On-From List)	193
HELP (Obtain RACF Help)	198
LISTDSD (List Data Set Profile)	200
LISTGRP (List Group Profile)	214
LISTUSER (List User Profile)	223
PASSWORD (Specify User Password)	242
PERMIT (Maintain Resource Access Lists)	247
RACDCERT (RACF Digital Certificate)	258
RACLINK (Administer User ID Associations)	296
RALTER (Alter General Resource Profile)	303
RDEFINE (Define General Resource Profile)	337
RDELETE (Delete General Resource Profile)	371
REMOVE (Remove User from Group)	376
RESTART (Restart RRSF Functions)	379
RLIST (List General Resource Profile)	382
RVARY (Change Status of RACF Database)	399
SEARCH (Search RACF Database)	408
SET	420
SETROPTS (Set RACF Options)	435
SIGNOFF (Sign Off Sessions)	476
STOP (Shutdown RRSF)	479
TARGET (Define RRSF Nodes)	481
 Appendix A. Naming Considerations for Resource Profiles	 495
Profile Definitions	495
Discrete Profiles	495
Generic Profiles	495
Fully-Qualified Generic Profiles (DATASET class only)	495
Determining RACF Protection	496
Profile Names for Data Sets	496
Discrete Profiles	497
Generic Profile Rules—Enhanced Generic Naming Inactive	497
Generic Profile Rules—Enhanced Generic Naming Active	498
Choosing between Discrete and Generic Profiles	500
Profile Names for General Resources	501
Permitting Profiles for GENERICOWNER Classes	503
Commands to Administer VM Shared File System Profiles	504
 Appendix B. Description of RACF Classes	 507
Describing RACF classes	507
Supplied resource classes for z/OS and OS/390 systems	507
Supplied resource classes for z/VM and VM systems	513
 Appendix C. Accessibility	 517
Using assistive technologies	517
Keyboard navigation of the user interface.	517
 Notices	 519
Trademarks.	520

Index	523
------------------------	------------

Figures

1.	Key to symbols in command syntax diagrams	10
2.	Sample ISPF Panel for RACF	17
3.	Example 1: Output for the DISPLAY Command	196
4.	Example 2: Output for the DISPLAY Command	196
5.	Example 3: Output for the DISPLAY Command	196
6.	Example 4: Output for the DISPLAY Command	197
7.	Example 5: Output for the DISPLAY Command	197
8.	Example 1: Output for the LISTDSD Command	210
9.	Example 2: Output for the LISTDSD Command	212
10.	Example 3: Output for the LISTDSD Command	212
11.	Example 4: Output for the LISTDSD Command	213
12.	Example 1: Output for LISTGRP RESEARCH.	219
13.	Example 2: Output for LISTGRP *	220
14.	Example 3: Output for LISTGRP DFPADMIN DFP	221
15.	Example 4: Output for LISTGRP DFPADMIN DFP NORACF	221
16.	Example 5: Output for LISTGRP OMVSG1 OMVS NORACF	221
17.	Example 6: Output for LISTGRP NETGROUP.	222
18.	Example 1: Output for LISTUSER	235
19.	Example 2: Output for LISTUSER (IBMUSER CALTMAN DAF0)	235
20.	Example 3: Output for LISTUSER DAF0 TSO.	237
21.	Example 4: Output for LISTUSER NORACF TSO	237
22.	Example 5: Output for LISTUSER DAF0 DFP.	238
23.	Example 6: Output for LISTUSER DAF0 NORACF DFP	238
24.	Example 7: Output for LISTUSER DAF0 NORACF CICS (prior to CICS 4.1)	238
25.	Example 8: Output for LISTUSER DAF0 NORACF CICS (CICS 4.1)	239
26.	Example 9: Output for LISTUSER DAF0 NORACF LANGUAGE	239
27.	Example 10: Output for LISTUSER DAF0 NORACF OPERPARM	239
28.	Example 11: Output for listing OMVS user information.	240
29.	Example 12: Output for LISTUSER CSMITH OMVS NORACF (Using Defaults)	240
30.	Example 13: Output for LISTUSER CSMITH NORACF DCE	240
31.	Example 14: Output for LISTUSER RONTOMS NORACF KERB.	240
32.	Example 15: Output for LISTUSER MRSERVER PROXY NORACF.	241
33.	Example 16: Output for LISTUSER MRSERVER EIM NORACF	241
34.	Controlling Access to RACDCERT Functions	288
35.	Example 2: Output for the RACDCERT LIST Command	291
36.	Example 4: Output for the RACDCERT CHECKCERT Command	292
37.	Example 5: Output from the LIST Command	293
38.	Example 6: Output from the LISTRING Command	294
39.	Example 12: Output from the LISTMAP Command	294
40.	Example 13: Output from the LISTMAP LABEL Command	294
41.	Example 16: Output from the LIST Command.	295
42.	Example 1: Output for the RACLINK LIST Command	302
43.	Example 1: Output for the RLIST Command	395
44.	Example 2: Output for the RLIST Command	396
45.	Example 3: Output for the RLIST Command with RESGROUP Option.	396
46.	Example 4: Output for RLIST Command with Masked Application Key.	397
47.	Example 5: Output for RLIST Command with Encrypted Application Key	397
48.	Example 6: Output for RLIST Command for STDATA Segment	397
49.	Example 7: Output for RLIST Command for KERB Segment	397
50.	Example 8: Output for RLIST Command for TSOIM13.	398
51.	Example 9: Output for RLIST FACILITY IRR.PROXY.DEFAULTS PROXY NORACF.	398
52.	Example 10: Output for RLIST FACILITY IRR.PROXY.DEFAULTS EIM PROXY NORACF	398
53.	Example 1: Output for the RVARY LIST Command	405

54.	Example 2: Output following Deactivation and Deallocation of RACF.PRIM1	406
55.	Example 3: Output following the Activation of RACF.BACK1	406
56.	Example 4: Output following the RVARY SWITCH,DATASET(RACF.PRIM1) Command	406
57.	Example 5: Output following the RVARY NODATASHARE Command	406
58.	Example 6: Output following the RVARY DATASHARE Command	407
59.	Example 3: Output for the SET LIST Command	433
60.	EXAMPLE 6: Response from the SET TRACE command for PDCALLABLE service.	433
61.	EXAMPLE 7: Output for the SET LIST Command for PDCALLABLE service.	434
62.	Output for Example 3: SETROPTS LIST.	474
63.	Example 1: Output for TARGET Command.	492

Tables

1. Functions of RACF Commands	2
2. How the RACF commands can be issued	8
3. How the ADDGROUP Command Can be Issued	24
4. ADDGROUP Examples	31
5. How the ADDSD Command Can be Issued	33
6. Examples of ADDSD	45
7. How the ADDUSER Command Can be Issued	48
8. ADDUSER Examples	85
9. How the ALTDSD Command Can be Issued	90
10. ALTDSD Examples	102
11. How the ALTGROUP Command Can be Issued	104
12. ALTGROUP Examples	113
13. How the ALTUSER Command Can be Issued	115
14. ALTUSER Examples	169
15. How the CONNECT Command Can be Issued	173
16. CONNECT Examples.	179
17. How the DELDSD Command Can be Issued	181
18. DELDSD Examples	185
19. How the DELGROUP Command Can be Issued	186
20. DELGROUP Example	188
21. How the DELUSER Command Can be Issued	190
22. DELUSER Examples	192
23. How the DISPLAY Command Can be Issued	193
24. DISPLAY Examples	195
25. How the HELP Command Can be Issued	198
26. HELP Examples	199
27. How the LISTDSD Command Can be Issued	202
28. LISTDSD Examples	209
29. How the LISTGRP Command Can be Issued	215
30. LISTGRP Examples	218
31. How the LISTUSER Command Can be Issued	224
32. LISTUSER Examples.	231
33. How the PASSWORD Command Can be Issued.	242
34. PASSWORD Examples	245
35. How the PERMIT Command Can be Issued	248
36. PERMIT Examples.	256
37. How the RACDCERT Command Can be Issued	258
38. RACDCERT Authority Checks	259
39. subjectKeyIdentifier Extension Logic for GENCERT.	271
40. authorityKeyIdentifier Extension Logic for GENCERT	271
41. Authority Required To Generate a Certificate	272
42. keyUsage Extension Logic for GENCERT	276
43. basicConstraints Extension Logic for GENCERT	276
44. subjectAltName Extension Logic for GENCERT	277
45. issuerAltName Extension Logic for GENCERT	277
46. Authority Required to Export a Certificate Package	277
47. Authority Required to Connect to One's Own Key Ring	279
48. Authority Required To Connect to Someone Else's Key Ring	279
49. RACDCERT Examples	289
50. How the RACLINK Command Can be Issued	296
51. RACLINK Examples	301
52. How the RALTER Command Can be Issued	303
53. RALTER Examples	334

54. How the RDEFINE Command Can be Issued	337
55. RDEFINE Examples	368
56. How the RDELETE Command Can be Issued.	371
57. RDELETE Examples	374
58. How the REMOVE Command Can be Issued	376
59. REMOVE Examples	378
60. How the RESTART Command Can be Issued.	379
61. RESTART Examples	381
62. How the RLIST Command Can be Issued	382
63. RLIST Examples	393
64. How the RVARY Command Can be Issued.	400
65. RVARY Examples	405
66. How the SEARCH Command Can be Issued	408
67. SEARCH Examples	418
68. How the SET Command Can be Issued	420
69. SET Examples	432
70. How the SETROPTS Command Can be Issued	437
71. SETROPTS Examples	471
72. How the SIGNOFF Command Can be Issued.	476
73. SIGNOFF Examples	478
74. How the STOP Command Can be Issued	479
75. STOP Example	480
76. How the TARGET Command Can be Issued	481
77. TARGET Examples	491
78. Generic Naming for Data Sets	496
79. Generic Naming for Data Sets with Enhanced Generic Naming Inactive—Asterisk at the End	497
80. Generic Naming for Data Sets with Enhanced Generic Naming Inactive—Asterisk in the Middle or %	498
81. Generic Data Set Profile Names Created with Enhanced Generic Naming Active—Asterisk and Double Asterisk at the End.	499
82. Generic Data Set Profile Names Created with Enhanced Generic Naming Active—Asterisk, Double Asterisk, or Percent Sign in the Middle	499
83. After Deactivating EGN—Asterisk and Percent Sign in the Middle	500
84. After Deactivating EGN—Asterisk and Double Asterisk at the End	500
85. Generic Naming for General Resources	501
86. Generic Naming for General Resources—Percent Sign, Asterisk, or Double Asterisk at the Beginning	503
87. Generic Naming for General Resources—Asterisk or Double Asterisk at the Ending.	503
88. Generic Naming for General Resources—Asterisk, Double Asterisk, or Percent Sign in the Middle	503
89. Permitting Profiles	504
90. Permitting Profiles (continued)	504
91. Commands to Administer SFS Profiles	504
92. Resource Classes for z/OS and OS/390 Systems	507
93. Resource Classes for z/VM and VM Systems	513

About this document

This document contains information about the Security Server for z/OS and z/OS.e, which consists of these components:

- Resource Access Control Facility (RACF)
- DCE Security Server
- z/OS Firewall Technologies
- Lightweight Directory Access Protocol (LDAP) Server, which includes client and server function
- Open Cryptographic Enhanced Plug-ins
- Security Server Network Authentication Service
- PKI Services

For information about the other components, see the publications related to those components.

Purpose of this document

This publication describes the syntax and the functions of the commands for RACF. The commands are presented in alphabetical order, and the operands within each command description are presented alphabetically. Exceptions occur where operands are positional, where alternative operands are grouped together or wherever alternative operand grouping is more practical for easier understanding.

The appendixes of this document contain information on generic and discrete profiles for data sets and general resources, as well as a list of the RACF® classes.

Who should use this document

This document is intended for RACF-defined users who are responsible for creating, updating, or maintaining the profiles for users, groups, data sets, and general resources in the RACF database.

Readers must be familiar with the RACF concepts and terminology. Many RACF functions also require you to understand the more detailed descriptions in *z/OS Security Server RACF Security Administrator's Guide*.

How to use this document

- If you want a concise list of all the RACF commands, see Chapter 1, "Introduction" on page 1.
- If you need a general discussion on entering RACF commands, see Chapter 2, "Basic information for issuing RACF commands" on page 7.
- If you need information on how to read syntax diagrams, see "Syntax of RACF commands and operands" on page 9.
- If you want information about entering a RACF command as a RACF TSO command, see Chapter 3, "RACF TSO commands" on page 15.
- If you want information about entering a RACF command as a RACF operator command, see Chapter 4, "RACF operator commands" on page 21.
- If you know the command you wish to enter, but are unsure of the syntax, see the chapter that documents the appropriate command.

Where to find more information

Where necessary, this document references information in other publications. For complete titles and order numbers for all elements of z/OS™, see *z/OS Information Roadmap*.

Softcopy publications

The RACF library is available on the following CD-ROMs. The CD-ROM online library collections include Library Reader™, which is a program that enables you to view the softcopy documents.

SK3T-4269 *z/OS Version 1 Release 4 Collection*

This collection contains the set of unlicensed documents for the current release of z/OS in both BookManager® and Portable Document Format (PDF) files. You can view or print the PDF files with the Adobe Acrobat reader.

SK3T-4272 *z/OS Security Server RACF Collection*

This softcopy collection kit contains the Security Server library for z/OS in both BookManager and Portable Document Format (PDF) files. You can view or print the PDF files with the Adobe Acrobat reader.

SK2T-2180 *Online Library OS/390 Security Server RACF Information Package*

This softcopy collection contains the Security Server library for OS/390. It also contains the RACF/MVS Version 2 product libraries, the RACF/VM 1.10 product library, product documents from the OS/390® and VM collections, International Technical Support Organization (ITSO) documents (known as Redbooks™), and Washington System Center (WSC) documents (known as orange books) that contain information related to RACF. The collection does not contain any licensed publications. By using this CD-ROM, you have access to RACF-related information from IBM products such as OS/390, VM/ESA®, CICS®, and NetView®.

SK3T-7876 *IBM eServer zSeries™ Redbooks Collection*

This softcopy collection contains a set of documents called Redbooks that pertain to zSeries subject areas ranging from e-business application development and enablement to hardware, networking, Linux, solutions, security, Parallel Sysplex® and many others.

SK2T-2177 *IBM Redbooks S/390® Collection*

This softcopy collection contains a set of documents called Redbooks that pertain to S/390 subject areas ranging from application development and enablement to hardware, networking, security, Parallel Sysplex and many others.

RACF courses

The following RACF classroom courses are available:

ES840 *Implementing RACF Security for CICS/ESA® and CICS/TS*

H3917 *Basics of OS/390 Security Server RACF Administration*

H3927 *Effective RACF Administration*

ES88A *Exploiting the Features of OS/390 Security Server RACF*

IBM provides a variety of educational offerings for RACF. For more information about classroom courses and other offerings, do any of the following:

- See your IBM representative
- Call 1-800-IBM-TEACH (1-800-426-8322)

Using LookAt to look up message explanations

LookAt is an online facility that allows you to look up explanations for most messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can access LookAt from the Internet at:

<http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/>

or from anywhere in z/OS where you can access a TSO/E command line (for example, TSO/E prompt, ISPF, z/OS UNIX System Services running OMVS). You can also download code from the *z/OS Collection* (SK3T-4269) and the LookAt Web site that will allow you to access LookAt from a handheld computer (Palm Pilot VIIx suggested).

To use LookAt as a TSO/E command, you must have LookAt installed on your host system. You can obtain the LookAt code for TSO/E from a disk on your *z/OS Collection* (SK3T-4269) or from the **News** section on the LookAt Web site.

Some messages have information in more than one document. For those messages, LookAt displays a list of documents in which the message appears.

Accessing z/OS licensed documents on the Internet

z/OS licensed documentation is available on the Internet in PDF format at the IBM Resource Link™ Web site at:

<http://www.ibm.com/servers/resourceLink>

Licensed documents are available only to customers with a z/OS license. Access to these documents requires an IBM Resource Link user ID and password, and a key code. With your z/OS order you received a Memo to Licensees, (GI10-0671), that includes this key code.¹

To obtain your IBM Resource Link user ID and password, log on to:

<http://www.ibm.com/servers/resourceLink>

To register for access to the z/OS licensed documents:

1. Sign in to Resource Link using your Resource Link user ID and password.
2. Select **User Profiles** located on the left-hand navigation bar.

Note: You cannot access the z/OS licensed documents unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

Printed licensed documents are not available from IBM.

1. z/OS.e™ customers received a Memo to Licensees, (GI10-0684) that includes this key code.

Preface

You can use the PDF format on either **z/OS Licensed Product Library CD-ROM** or IBM Resource Link to print licensed documents.

IBM systems center publications

IBM systems centers produce documents known as red and orange books that can help you set up and use RACF. These documents have not been subjected to any formal review nor have they been checked for technical accuracy, but they represent current product understanding (at the time of their publication) and provide valuable information on a wide range of RACF topics. They are not shipped with RACF; you must order them separately. A selected list of these documents follows. Other documents are available, but they are not included in this list, either because the information they present has been incorporated into IBM product manuals or because their technical content is outdated.

G320-9279	<i>Systems Security Publications Bibliography</i>
GG22-9396	<i>Tutorial: Options for Tuning RACF</i>
GG24-3378	<i>DFSMS and RACF Usage Considerations</i>
GG24-3451	<i>Introduction to System and Network Security: Considerations, Options, and Techniques</i>
GG24-3524	<i>Network Security Involving the NetView Family of Products</i>
GG24-3970	<i>Elements of Security: RACF Overview - Student Notes</i>
GG24-3971	<i>Elements of Security: RACF Installation - Student Notes</i>
GG24-3972	<i>Elements of Security: RACF Advanced Topics - Student Notes</i>
GG24-3984	<i>RACF Macros and Exit Coding</i>
GG24-4282	<i>Secured Single Signon in a Client/Server Environment</i>
GG24-4453	<i>Enhanced Auditing Using the RACF SMF Data Unload Utility</i>
GG26-2005	<i>RACF Support for Open Systems Technical Presentation Guide</i>
GC28-1210	<i>System/390® MVS™ Sysplex Hardware and Software Migration</i>
SG24-4704	<i>OS/390 Security Services and RACF-DCE Interoperation</i>
SG24-4820	<i>OS/390 Security Server Audit Tool and Report Application</i>
SG24-5158	<i>Ready for e-business: OS/390 Security Server Enhancements</i>
SG24-5339	<i>The OS/390 Security Server Meets Tivoli®: Managing RACF with Tivoli Security Products</i>

Other sources of information

IBM provides customer-accessible discussion areas where RACF may be discussed by customer and IBM participants. Other information is also available through the Internet.

IBM discussion areas

| IBM provides *ibm.servers.mvs.racf* newsgroup for discussion of RACF-related
| topics. You can find this newsgroup on news (NNTP) server *news.software.ibm.com*
| using your favorite news reader client.

Internet sources

The following resources are available through the Internet to provide additional information about the RACF library and other security-related topics:

- **Online library**

To view and print online versions of the z/OS publications, use this address:

<http://www.ibm.com/servers/eserver/zseries/zos/bkserv/>

- **Redbooks**

The documents known as Redbooks that are produced by the International Technical Support Organization (ITSO) are available at the following address:

<http://www.ibm.com/redbooks/>

- **Enterprise systems security**

For more information about security on the S/390 platform, OS/390, and z/OS, including the elements that comprise the Security Server, use this address:

<http://www.ibm.com/servers/eserver/zseries/zos/security/>

- **RACF home page**

You can visit the RACF home page on the World Wide Web using this address:

<http://www.ibm.com/servers/eserver/zseries/zos/racf/>

- **RACF-L discussion list**

Customers and IBM participants may also discuss RACF on the RACF-L discussion list. RACF-L is not operated or sponsored by IBM; it is run by the University of Georgia.

To subscribe to the RACF-L discussion and receive postings, send a note to:

listserv@listserv.uga.edu

Include the following line in the body of the note, substituting your first name and last name as indicated:

`subscribe racf-l first_name last_name`

To post a question or response to RACF-L, send a note, including an appropriate Subject: line, to:

racf-l@listserv.uga.edu

- **Sample code**

You can get sample code, internally-developed tools, and exits to help you use RACF. This code works in our environment, at the time we make it available, but is not officially supported. Each tool or sample has a README file that describes the tool or sample and any restrictions on its use.

To access this code from a Web browser, go to the RACF home page and select the “Downloads” topic from the navigation bar, or go to <ftp://ftp.software.ibm.com/eserver/zseries/zos/racf/>.

The code is also available from [ftp.software.ibm.com](ftp://ftp.software.ibm.com) through anonymous FTP. To get access:

1. Log in as user **anonymous**.
2. Change the directory, as follows, to find the subdirectories that contain the sample code or tool you want to download:

```
cd eserver/zseries/zos/racf/
```

An announcement will be posted on the RACF-L discussion list and on newsgroup *ibm.servers.mvs.racf* whenever something is added.

Note: Some Web browsers and some FTP clients (especially those using a graphical interface) might have problems using [ftp.software.ibm.com](ftp://ftp.software.ibm.com) because of inconsistencies in the way they implement the FTP protocols. If you have problems, you can try the following:

- Try to get access by using a Web browser and the links from the RACF home page.
- Use a different FTP client. If necessary, use a client that is based on command line interfaces instead of graphical interfaces.

Preface

- If your FTP client has configuration parameters for the type of remote system, configure it as UNIX[®] instead of MVS.

Restrictions

Because the sample code and tools are not officially supported,

- There are no guaranteed enhancements.
- No APARs can be accepted.

To request copies of IBM publications

Direct your request for copies of any IBM publication to your IBM representative or to the IBM branch office serving your locality.

There is also a toll-free customer support number (1-800-879-2755) available Monday through Friday from 6:30 a.m. through 5:00 p.m. Mountain Time. You can use this number to:

- Order or inquire about IBM publications
- Resolve any software manufacturing or delivery concerns
- Activate the program reorder form to provide faster and more convenient ordering of software updates

Summary of changes

Summary of changes for SA22-7687-03 z/OS Version 1 Release 4

This document contains information previously presented in *z/OS Security Server RACF Command Language Reference*, SA22-7687-02, which supports z/OS Version 1 Release 3.

New information

- Information has been added to indicate this document supports z/OS.e.
- A new PCICC sub-operand has been added to the GENCERT operand in the RACDCERT command.
- A new EIM operand has been added to the ADDUSER, ALTUSER, LISTUSER, RALTER, RDEFINE, RLIST and SETROPTS commands.
- Additional UID/GID support has been added to the ADDGROUP, ADDUSER, ALTGROUP, ALTUSER and SEARCH commands.
- New PADS support has been added to RALTER and RDEFINE.
- Additional PKI services support has been added to the RACDCERT command.
- The SETROPTS command has been updated to support Multi-level security (MLS).

Changed information

- The descriptions of the RACDCERT operands have been restructured for clarity.
- The description of APPLDATA for the RDEFINE and RALTER commands has been reorganized for clarity.

This document includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Starting with z/OS V1R2, you may notice changes in the style and structure of some content in this document—for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our documents.

Summary of changes for SA22-7687-02 z/OS Version 1 Release 3

This document contains information previously presented in *z/OS SecureWay Security Server RACF Command Language Reference*, SA22-7687-01, which supports z/OS Version 1 Release 2.

New information

- New information has been added to the ALTUSER command describing the results if an unauthorized user specifies the UAUDIT keyword.
- The new PROXY keyword in support of Policy Director is now provided for the ADDUSER, ALTUSER, LISTUSER, RALTER, RDEFINE and RLIST commands.

- The SET command now supports the new PDCALLABLE keyword for the TRACE operand in support of IBM Policy Director.
- The SET command Callable Services table now supports IRRSCL00 for Access Control Lists.

Changed information

- The descriptions of the RESTRICTED keyword has been modified for z/OS UNIX support in the ADDUSER and ALTUSER commands.
- The description of MAXSIZE and SIZE subkeywords to the TSO keyword of the ADDUSER and ALTUSER commands has been modified to show their relationship to each other.
- The RDELETE command definition of *profile-name* has been enhanced regarding profiles in the CACHECLS class in support of IBM Policy Director.
- The Callable Services Table for the SET command has been modified to identify the callable services correctly.

Deleted information

- The RACF glossary has been removed from this publication. See *z/OS Security Server RACF Security Administrator's Guide* for the RACF glossary.

This document includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Summary of changes for SA22-7687-01 z/OS Version 1 Release 2

This document contains information previously presented in *z/OS SecureWay Security Server RACF Command Language Reference*, SA22-7687-00, which supports z/OS Version 1 Release 1.

New information

- The ADDUSER command supports the following for the keyword KERB:
 - ENCRYPT and the following subkeywords:
 - DES | NODES
 - DES3 | NODES3
 - DESD | NODESD
- The ADDGROUP command supports the following keywords:
 - UNIVERSAL
- The ALTUSER command supports the following for the keyword KERB:
 - ENCRYPT and the following subkeywords:
 - DES | NODES
 - DES3 | NODES3
 - DESD | NODESD
 - NOENCRYPT
- The LISTGRP command displays the new attribute:
 - UNIVERSAL
- The LISTUSER command displays new information for the KERB attribute:
 - KEY ENCRYPTION TYPE=DES DES3 DESD

- The RALTER command supports the following for the keyword KERB:
 - ENCRYPT and the following subkeywords:
 - DES | NODES
 - DES3 | NODES3
 - DESD | NODESD
 - NOENCRYPT
- The RDEFINE command supports the following for the keyword KERB:
 - ENCRYPT and the following subkeywords:
 - DES | NODES
 - DES3 | NODES3
 - DESD | NODESD
- The RLIST command displays new information for the KERB attribute:
 - KEY ENCRYPTION TYPE=DES DES3 DESD
- The SET command supports the following new keywords for the TRACE operand:
 - [ASID(*asid* ... |*) | NOASID | ALLASIDS]
 - [CALLABLE(ALL | NONE | TYPE(*type* ...)) | NOCALLABLE]
 - [DATABASE([ALL | NONE] [ALTER | NOALTER] [ALTERI | NOALTERI] [READ | NOREAD]) | NODATABASE]
 - [JOBNAME(*jobname* ... |*) | NOJOBNAME | ALLJOBNAMES]
 - [RACROUTE(ALL | NONE | TYPE(*type* ...)) | NORACROUTE]
- The SETROPTS command supports the following new keyword:
 - KERBLVL(0 | 1)

Changed information

- The RDEFINE, RALTER, RLIST, RDELETE and PERMIT commands have been enhanced to accept mixed case profile names for classes whose CDT entries are specified with CASE=ASIS.
- The RDEFINE and RALTER commands have been enhanced to accept mixed case member names in the ADDMEM and DELMEM operands for mixed case classes.
- The ADDSD, RDEFINE and PERMIT commands have been enhanced to accept mixed case profile names in the FROM operand when the FCLASS operand specifies a mixed case class.
- The SEARCH command has been enhanced to support mixed case strings on the FILTER and MASK operands for mixed case classes, and on the CLIST operand for any class.

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.

Chapter 1. Introduction

The profiles in the RACF database contain the information RACF needs to control access to resources. The RACF commands allow you to add, change, delete, and list the profiles for:

- Users
- Groups
- Data sets
- General resources, which include terminals, DASD volumes, and all other resource classes defined in the RACF class descriptor table (CDT)

Table 1 shows, in alphabetic order, each command, and its function.

Most RACF functions do not require special versions or releases of the operating system or operating system components. Some, however, do require that your system be at a certain level. If you are unsure about whether a particular RACF function is available with your system, see your security administrator.

Some commands require that the RACF subsystem be active or that you have authorization to issue the commands. Refer to the “Authorization Required” section with each command for details on the authorization required.

The following RACF commands are available only on RACF for VM:

- ADDFILE
- ADDDIR
- ALTFILE
- ALTDIR
- DELFILE
- DELDIR
- LFILE
- LDIRECT
- PERMFILE
- PERMDIR
- SRFILE
- SRDIR

See the *RACF Command Language Reference* for your VM system for more information.

Summary of commands and their functions

RACF commands allow you to list, modify, add, and delete profiles for users, groups, connect entries, and resources. Table 1 on page 2 shows, in alphabetic order, each of the commands and its functions.

Introduction

Table 1. Functions of RACF Commands

RACF Command	Command Functions
ADDGROUP	<ul style="list-style-type: none">• Define one or more new groups as a subgroup of an existing group.• Specify a model data set profile for a group.• Define default DFP information for a group.• Define the z/OS UNIX information for a group.• Define a group as a universal group.
ADDSD	<ul style="list-style-type: none">• RACF-protect one or more existing data sets.• RACF-define one or more data sets brought from another system where they were RACF-protected.• RACF-define generic data set profiles.• Create a new data set model profile.
ADDUSER	<ul style="list-style-type: none">• Define one or more new users and connect the users to their default connect group.• Specify a model data set profile for a user.• Define CICS operator information.• Define default DFP information for a user.• Define the EIM information for a user.• Define the preferred national language.• Define default operator information.• Define default TSO logon information for a user.• Define default work attributes.• Define the z/OS UNIX information for a user.• Define the DCE information for a user.• Define default NetView operator information.• Define the COMMAND field of the logon panel.• Define the LNOTES and NDS information for a user.• Define access checking with the RESTRICTED and NORESTRICTED keywords.• Define the KERB information for a user.• Define the PROXY information for a user.
ALTDSD	<ul style="list-style-type: none">• Change one or more discrete or generic data set profiles.• Protect a single volume of a multivolume, non-VSAM DASD data set.• Remove protection from a single volume of a multivolume, non-VSAM DASD data set.
ALTGROUP	<ul style="list-style-type: none">• Change the information in one or more group profiles (such as the superior group, owner, or model profile name).• Change or delete the default DFP information for a group.• Add, change, or delete the information for the z/OS UNIX group.
ALTUSER	<ul style="list-style-type: none">• Change the information in one or more user profiles (such as the owner, universal access authority, or security level).• Revoke or reestablish one or more users' privileges to access the system.• Specify logging of information about the user, such as the commands the user issues.• Change or delete CICS operator information.• Change or delete the default DFP information for a user.• Define, change or delete the EIM information for a user.• Change the preferred national language.• Change or delete the default operator information.• Change or delete the default TSO logon information for a user.• Change or delete the default work attributes.• Add, change, or delete the information for the z/OS UNIX user.• Change the DCE information for a user.• Change or delete NetView operator information.• Manipulate the COMMAND field of the logon panel.• Change the LNOTES and NDS information for a user.• Change access checking methods with the RESTRICTED and NORESTRICTED keywords.• Add or change the KERB information for a user.• Add or change the PROXY information for a user.
CONNECT	<ul style="list-style-type: none">• Connect one or more users to a group.• Modify one or more users' connection to a group.• Revoke or reestablish one or more users' privileges to access the system.
DELDSD	<ul style="list-style-type: none">• Delete one or more discrete or generic data set profiles.• Delete a discrete data set profile for a tape data set, while retaining the data set name in the TVTOC.• Remove a data set profile, but leave the data set RACF-indicated, when moving a RACF-protected data set to another system that has RACF.
DELGROUP	<ul style="list-style-type: none">• Delete one or more groups and their relationship to the superior group.

Table 1. Functions of RACF Commands (continued)

RACF Command	Command Functions
DELUSER	<ul style="list-style-type: none"> Delete one or more users and remove all of their connections to RACF groups.
DISPLAY	<ul style="list-style-type: none"> Display users signed on to a RACF subsystem.
HELP	<ul style="list-style-type: none"> Display the function and proper syntax of RACF commands.
LISTDSD	<ul style="list-style-type: none"> List the details of one or more discrete or generic data set profiles, including the users and groups authorized to access the data sets. Determine the most specific matching generic profile for a data set. Perform a local refresh of generic DATASET profiles.
LISTGRP	<ul style="list-style-type: none"> List the details of one or more group profiles, including the users connected to the group. List only the information contained in a specific segment (RACF, DFP, or OMVS) of the group profile. Display limited information if the group is a UNIVERSAL group.
LISTUSER	<ul style="list-style-type: none"> List the details of one or more user profiles, including all of the groups to which each user is connected. List only the information contained in a specific segment (for example, DFP or OMVS information) of a user profile.
PASSWORD	<ul style="list-style-type: none"> Change one or more users' passwords. Change one or more users' password change interval. Reset one or more users' passwords to a known default value.
PERMIT	<ul style="list-style-type: none"> Give or remove authority to access a resource to specific users or groups. Change the level of access authority to a resource for specific users or groups. Copy the list of authorized users from one resource profile to another. Delete an existing access list.
RACDCERT	<ul style="list-style-type: none"> List information about the existing certificate definitions for a specified RACF-defined user ID or the requester's user ID. Add a certificate definition and associate it with a specified RACF-defined user ID or the requester's user ID and set the TRUST flag. Check to see if a certificate has been defined to RACF. Alter the TRUST flag for an existing definition. Delete an existing definition. Add or remove a certificate to a key ring. Create, delete, or list an existing key ring. Generate a public/private key pair and certificate. Write a certificate or certificate package to a data set. Create a certificate request. Create, alter, delete, or list a certificate name filter. Gather diagnostic information.
RACLINK	<ul style="list-style-type: none"> Define, approve, and delete (undefine) a user ID association. List information related to a user ID association. Establish password synchronization between user IDs.
RALTER	<ul style="list-style-type: none"> Change the discrete or generic profiles for one or more resources whose class is defined in the class descriptor table. Maintain the global access checking tables. Maintain security category and security level tables. List the encryption keys used if the profile has a KERB segment. Maintain DLFDATA, SESSION, SSIGNON, and STDATA segment information in the profiles. Define, change, or delete the DOMAINDN, OPTIONS and LOCALREGISTRY information in the EIM segment. Change profiles associated with a SystemView for MVS application. Define, change, or delete the LDAPHOST address, BINDDN and BINDPW information in the PROXY segment.

Introduction

Table 1. Functions of RACF Commands (continued)

RACF Command	Command Functions
RDEFINE	<ul style="list-style-type: none"> • RACF-protect by a discrete or generic profile any resource whose class is defined in the class descriptor table. • Define entries in the global access checking tables. • Define security category and security level tables. • Define the encryption keys used if the profile has a KERB segment. • Define DLFDATA, SESSION, SSIGNON, and STDATA segment information in the profiles. • Define the EIM segment and the DOMAINDN, OPTIONS and LOCALREGISTRY information for the segment. • Define the list of classes for which RACF is to save RACLISTed results on the RACF database. • Define profiles associated with a SystemView for MVS application. • Create, alter, or delete additional criteria for a certificate name filter. • Define the LDAPHOST address, BINDDN and BINDPW information if the profile has a PROXY segment. • Define the SHARED.IDS profile in the UNIXPRIV class, which controls the default behavior of adding or altering UIDs and GIDs in the OMVS segment. • Modify the processing of z/OS UNIX System Services by defining profiles in the UNIXPRIV class that either: <ul style="list-style-type: none"> – Serve as system-wide options (such as the CHOWN.UNRESTRICTED or FILE.GROUPOWNER.SETGID profiles). – Define an individual superuser capability (such as the ability to change file ownership using the SUPERUSER.FILESYS.CHOWN resource), which can be granted to a user instead of assigning UID(0).
RDELETE	<ul style="list-style-type: none"> • Remove RACF-protection from one or more resources whose class is defined in the class descriptor table. • Delete the global access checking tables. • Delete the security category and security level tables. • Delete a class from the list of classes for which RACF saves RACLISTed results on the RACF database.
REMOVE	<ul style="list-style-type: none"> • Remove one or more users from a group and assign a new owner for any group data sets owned by the users.
RESTART	<ul style="list-style-type: none"> • Restart a function in the RACF subsystem address space. • Restart the connection to a specific member system on a multisystem node.
RLIST	<ul style="list-style-type: none"> • List the details of discrete or generic profiles for one or more resources whose class is defined in the class descriptor table. • List the contents of the DLFDATA, SESSION, SSIGNON, and STDATA segments in the profiles. • List the DOMAINDN, OPTIONS and LOCALREGISTRY information if the profile has an EIM segment. • List the encryption keys used if the profile has a KERB segment. • List the LDAPHOST address, BINDDN information and whether or not a BINDPW exists if the profile has a PROXY segment. • Perform a local refresh of generic general resource profiles.
RVARY	<ul style="list-style-type: none"> • Dynamically deactivate and reactivate the RACF function. • Dynamically deactivate and reactivate the RACF primary and backup database. • Switch the primary and backup RACF databases. • Deactivate resource protection, for any resource whose class is defined in the class descriptor table, while RACF is deactivated. • Select operational mode when RACF is enabled for sysplex communication.

Table 1. Functions of RACF Commands (continued)

RACF Command	Command Functions
SEARCH	<ul style="list-style-type: none"> Obtain a list of RACF profile names that meet the search criteria for a class of, resources, users, or groups. These profile names can then be displayed on your terminal. <ul style="list-style-type: none"> Profile names that contain a specific character string Profiles for resources that have not been referenced for more than a specific number of days Profiles that RACF recognizes as model profiles Data set and general resource profiles that contain a level equal to or greater than the level you specify User and resource profiles that contain a security label that matches the security label you specify. User and resource profiles that contain a security level that matches the security level that you specify User and resource profiles that contain an access category that matches the access category that you specify. User profiles that contain an OMVS UID equal to the UID you specify. Group profiles that contain an OMVS GID equal to the GID you specify. Profiles for tape volumes that contain only data sets with an expiration date that matches the criteria you specify. Profiles for data sets that reside on specific volumes (or VSAM data sets that are cataloged in catalogs on specific volumes). Profiles for tape data sets, non-VSAM DASD data sets, or VSAM data sets. Format the selected profile names with specific character strings into a series of commands or messages and retain them in a CLIST data set. Create a CLIST of the RACF profile names that meet a search criteria for a class of resources.
SET	<ul style="list-style-type: none"> List information related to RACF remote sharing facility (RRSF) on the local node. Specify the name of a member of the RACF parameter library to be processed by RRSF. Set tracing on or off for specified operands. Specify options for automatic command direction.
SETROPTS	<p>Dynamically set system-wide options relating to resource protection, specifically:</p> <ul style="list-style-type: none"> Choose the resource classes that RACF is to protect. Gather and display RACF statistics. Set the universal access authority (UACC) for terminals. Set the KERBLVL to be used if a profile has a KERB segment. Specify logging of certain RACF commands and events. Permit list-of-groups access checking. Display options currently in effect. Enable or disable generic profile checking on either a class-by-class or system-wide level. Control user password syntax rules. Establish password syntax rules. Activate password processing for checking previous passwords, limit invalid password attempts, and warn of password expiration. Control global access checking for selected individual resources or generic names with selected generalized access rules. Set the passwords for authorizing use of the RVARV command. Initiate refreshing of in-storage generic profile lists and global access checking tables. Enable or disable shared generic profiles for general resources in common storage. Enable or disable shared profiles through RACLIST processing for general resources. Activate or deactivate auditing of access attempts to RACF-protected resources based on installation-defined security levels. Activate enhanced generic naming. Control the use of automatic data set protection (ADSP). Activate profile modeling for GDG, group, and user data sets. Activate protection for data sets with single-level names. Control logging of real data set names. Control the job entry subsystem (JES) options. Activate tape data set protection. Control whether or not data sets must be RACF-protected. Control the erasure of scratched DASD data sets. Activate program control. Control whether a profile creator's user ID is automatically added to the profile's access list. Make the name of the local RACF registry available to EIM services.
SIGNOFF	<ul style="list-style-type: none"> Sign off users from a RACF subsystem.
STOP	<ul style="list-style-type: none"> Stop the RACF subsystem address space.

Introduction

Table 1. Functions of RACF Commands (continued)

RACF Command	Command Functions
TARGET	<ul style="list-style-type: none">• List the controls and operational characteristics of the specified target RRSF nodes.• Specify the name of the target RRSF node.• Request an operational state for connection to the target RRSF node.• Delete an RRSF node from the local node.• Specify a description of the target RRSF node.• Purge the workspace data sets managed by RRSF in the RACF subsystem address space.• Specify the protocol type for the transport mechanism to be used in communication between the two RRSF nodes.• Specify the relationship between the target RRSF node and the node being configured.• Specify a prefix for the workspace data sets allocated by and used by RRSF for each target node.• Specify the characteristics of the workspace data sets associated with the node being defined to RRSF.• Specify the name of a multisystem node.• Identify the main system in a multisystem RRSF node.

Note: In data sharing mode or read-only mode, RACF employs global ENQs to serialize access to the RACF database before adding or removing protection from a resource. Otherwise—unless the installation has explicitly converted to GRS—RACF uses hardware RESERVE/RELEASE.

Chapter 2. Basic information for issuing RACF commands

You can use the RACF commands to add, modify, or delete RACF profiles and to define system-wide options. Before you can issue a RACF command, you must be defined to RACF with a sufficient level of authority.

How to enter RACF commands

There are several ways to enter RACF commands.

RACF TSO commands

Some RACF commands can be entered as RACF TSO commands. For information on entering RACF commands as RACF TSO commands, see Chapter 3, “RACF TSO commands” on page 15. For a complete list of which RACF commands can be entered as RACF TSO commands, see Table 2 on page 8.

RACF operator commands

Some RACF commands can be entered as RACF operator commands. For information on entering RACF commands as RACF operator commands, see Chapter 4, “RACF operator commands” on page 21. For a complete list of which RACF commands can be entered as RACF operator commands, see Table 2 on page 8.

Command direction and automatic command direction

With command direction, some RACF commands can be directed to run under the authority of a user ID on a remote node, or the same node. Use the AT keyword on your command for command direction. For information on command direction, see *z/OS Security Server RACF General User's Guide* or *z/OS Security Server RACF Security Administrator's Guide*. For information on the AT keywords, see the eligible command descriptions. For a complete list of RACF commands that are eligible for command direction, see Table 2 on page 8.

The possibility of failing while attempting to execute a command issued on one (up-level) system and manually or automatically directed to another (down-level) system through RACF remote sharing has been present since the introduction of RACF 2.2. This failure can occur for any of the following reasons:

- The command references a class unknown to the target system (for example, if the class descriptor tables are different),
- The command references a segment or field unknown to the target system (for example, if the templates or dynamic parse definition are different)
- The command uses a command keyword unknown to the target (for example, if the dynamic parse definitions or command processor code is different), or if it specifies a profile or member name that is unacceptable to the target system (for example, if the class descriptor tables have different syntax requirements for profile name length or syntax).

If an out-of-synchronization condition occurs while using automatic command direction, a RACF TSO command can be directed with the ONLYAT keyword to fix the condition. The command runs on the node specified on the ONLYAT keyword and are propagated to any other node. (Note that if the AT keyword is used, the command can be propagated by automatic command direction to other nodes.) For

Basic information

information on the ONLYAT keyword, see the eligible command descriptions. For a complete list of RACF commands that are eligible for automatic command direction, see Table 2.

Some RACF TSO commands can be automatically directed to remote nodes in order to keep profiles synchronized between the nodes. For information on automatic command direction, see *z/OS Security Server RACF Security Administrator's Guide*.

RACF parameter library

Some RACF commands can be processed from the RACF parameter library. For information on using the RACF parameter library, see *z/OS Security Server RACF System Programmer's Guide*. For a complete list of commands that can be processed from within the RACF parameter library, see Table 2.

R_admin callable service

You can also issue commands by calling the r_admin callable service (IRRSEQ00). For more information on using this callable service, and for a complete list of commands that can be issued in this manner, see *z/OS Security Server RACF Callable Services*.

Summary

Table 2 lists the ways you can enter each RACF command.

Table 2. How the RACF commands can be issued

RACF command	As a RACF TSO command?	As a RACF operator command?	With command direction?	With automatic command direction?	From the RACF parameter library?
ADDGROUP	Yes	Yes	Yes	Yes	Yes
ADDSD	Yes	Yes	Yes	Yes	Yes
ADDUSER	Yes	Yes	Yes	Yes	Yes
ALTDSD	Yes	Yes	Yes	Yes	Yes
ALTGROUP	Yes	Yes	Yes	Yes	Yes
ALTUSER	Yes	Yes	Yes	Yes	Yes
BLKUPD (See Note 1)	Yes	No	No	No	No
CONNECT	Yes	Yes	Yes	Yes	Yes
DELDSD	Yes	Yes	Yes	Yes	Yes
DELGROUP	Yes	Yes	Yes	Yes	Yes
DELUSER	Yes	Yes	Yes	Yes	Yes
DISPLAY	No	Yes	No	No	Yes
HELP	Yes	No	No	No	No
LISTDSD	Yes	Yes	Yes	No	Yes
LISTGRP	Yes	Yes	Yes	No	Yes
LISTUSER	Yes	Yes	Yes	No	Yes
PASSWORD	Yes	Yes	Yes	Yes	Yes
PERMIT	Yes	Yes	Yes	Yes	Yes

Table 2. How the RACF commands can be issued (continued)

RACF command	As a RACF TSO command?	As a RACF operator command?	With command direction?	With automatic command direction?	From the RACF parameter library?
RACDCERT	Yes	No	No	No	No
RACLINK	Yes	Yes	No	No	No
RALTER	Yes	Yes	Yes	Yes	Yes
RDEFINE	Yes	Yes	Yes	Yes	Yes
RDELETE	Yes	Yes	Yes	Yes	Yes
REMOVE	Yes	Yes	Yes	Yes	Yes
RESTART	No	Yes	No	No	No
RLIST	Yes	Yes	Yes	No	Yes
RVARY	Yes	Yes	No	No	Yes
SEARCH	Yes	Yes	Yes (See Note 2)	No	Yes
SET	No	Yes	No	No	Yes
SETOPTS	Yes	Yes	Yes	Yes (See Note 3)	Yes
SIGNOFF	No	Yes	No	No	Yes
STOP	No	Yes	No	No	No
TARGET	No	Yes	No	No	Yes

Notes:

1. Information about BLKUPD, the block update command, appears in *z/OS Security Server RACF Diagnosis Guide*, not this book.
2. SEARCH is not eligible for command direction when the CLIST keyword is specified.
3. SETOPTS LIST without other keywords is not eligible for automatic command direction.

Syntax of RACF commands and operands

This publication describes the syntax and function of the RACF commands. The commands are presented in alphabetical order. Each command description contains several examples.

For the key to the symbols used in the command syntax diagrams, see Figure 1 on page 10.

Basic information

1. UPPERCASE LETTERS or WORDS must be coded as they appear in the syntax diagrams but do not have to be uppercase.
2. Lowercase letters or words represent variables for which you must supply a value.
3. Parentheses () must be entered exactly as they appear in the syntax diagram.
4. An ellipsis ... (three consecutive periods) indicates that you can enter the preceding item more than once.
5. A single item in brackets [] indicates that the enclosed item is optional. Do not specify the brackets in your command.
6. Stacked items in brackets [] indicate that the enclosed items are optional. You can choose one or none. Do not specify the brackets in your command.
7. Stacked items in braces { } indicate that the enclosed items are alternatives. You must specify one of the items. Do not specify the braces in your command.

Note: When you select a bracket that contains braces, you must specify one of the alternatives enclosed within the braces.

8. Items separated by a vertical bar | indicate that you can specify only one of the items. Do not specify the vertical bar in your command.
9. An underlined operand indicates the default value when no alternate value is specified.
10. **BOLDFACE** or **boldface** indicates information that must be given for a command.
11. Single quotes ' ' indicate that information must be enclosed in single quotes.

Figure 1. Key to symbols in command syntax diagrams

The syntax for all occurrences of the *userid*, *group-name*, *password*, *class-name*, *profile-name*, *volume-serial*, *terminal-id*, and *date* operands in this book is as follows:

userid	<p>1-8 alphanumeric characters. The user ID can consist entirely of numbers and need not begin with any particular character.</p> <p>For TSO users who are defined to RACF, the user ID cannot exceed seven characters and must begin with an alphabetic, # (X'7B'), \$ (X'5B'), or @ (X'7C') character.</p>
group-name	<p>1-8 alphanumeric characters beginning with an alphabetic, # (X'7B'), \$ (X'5B'), or @ (X'7C') character. (You can set the default prefix to a group name only if the group name contains 1-7 characters. If the group name has 8 characters, you must always enter fully-qualified group data set names on the commands.</p>
password	<p>1-8 alphanumeric characters. Each installation can define its own password syntax rules.</p>
class-name	<p>Valid class names are USER, GROUP, DATASET, and those classes defined in the class descriptor table.</p>

The entries supplied by IBM in the class descriptor table are listed in Appendix B, “Description of RACF Classes” on page 507.

profile-name	Either a discrete name or a generic name, as described in Appendix A, “Naming Considerations for Resource Profiles” on page 495.
terminal-id	1-8 alphanumeric characters.
volume-serial	1-6 alphanumeric characters.
date	RACF interprets dates as 20yy when the year is less than 71, and 19yy when the year is greater than 71.

Return codes from RACF commands

All of the RACF commands (except RVARY) issue the following return codes. RVARY issues return codes of 0, 8, and 12.

Decimal Code	Meaning
0	Normal completion.
4	The command encountered a user error or an authorization failure and attempted to continue processing. Refer to documentation of the error message that RACF issues to determine what part of the current entity (if any) was completed. If additional entities were specified on the command, RACF attempts to process them.
8	The command encountered a user error or an authorization failure and terminated processing.
12	The command encountered a system error and terminated processing.

Except for commands entered using the AT keyword or the RACLINK command, you can use CLIST processing or REXX exec processing to check for these return codes. Commands entered using the AT keyword or the RACLINK command run in two phases:

- The first phase validates the issuer’s authority to use the function and determines whether the RACF subsystem address space is available to handle the second phase. A return code of 0 from the first phase means the request was successfully passed to the RACF subsystem address space. A return code of 8 means the command was rejected.
- The second phase runs in the RACF subsystem address space. Return codes within the address space cannot be interrogated by the issuer’s CLIST or REXX exec. The success of the command processing within the second phase must be determined from the returned output messages, if any. See *z/OS Security Server RACF General User’s Guide* for a description of the returned output.

RACF command restriction for non-base segments in RACF profiles

RACF has a limit of 255 on the number of operands that can be entered for a non-base profile segment on a single command. Examples of non-base segments in the general resource profile are:

- SESSION

Basic information

- STDATA
- TME

Examples of non-base segments in the user resource profile are:

- NETVIEW
- OMVS
- TSO

The RACF commands that can create or alter non-base segments in profiles, and that operate under this restriction, are:

- ADDGROUP
- ADDSD
- ADDUSER
- ALTDSD
- ALTGROUP
- ALTUSER
- RALTER
- RDEFINE

If a command includes more than 255 non-base segment operands for a non-base profile segment, the command will not work correctly. Only the remainder of $n/256$ operands will be applied to the database, where n is the total number of a particular segment's operands. The command will appear to have run successfully.

Installation exit routines from RACF commands

RACF provides a general purpose exit, IRREVX01, which can be modified by installations to receive control when most RACF TSO commands are issued.

ADDGROUP	ALTUSER	LISTDSD	PERMIT	REMOVE
ADDSD	CONNECT	LISTGRP	RALTER	RLIST
ADDUSER	DELDSD	LISTUSER	RDEFINE	SEARCH
ALTDSD	DELGROUP	PASSWORD	RDELETE	SETROPTS
ALTGROUP	DELUSER			

RACF does not invoke IRREVX01 for BLKUPD, RVARY, RACDCERT and RACLINK or for true RACF operator commands such as RESTART, TARGET, and SIGNOFF.

Your location might use installation-written exit routines to take additional security actions during RACF command processing, and these actions can affect the results you get when you issue a RACF command. For example, your location could use the ICHPWX01 preprocessing exit to install its own routine to examine a new password and new password interval.

For a complete description of RACF installation exits, see *z/OS Security Server RACF System Programmer's Guide*.

Attribute and authority summary

Each command description in this book includes a section called "Authorization Required," which describes how attributes and authorities affect your use of that command.

Group authorities

The group authorities, which define user responsibilities within the group, are shown below in order of least to most authority. Each level includes the privileges of the levels above it.

USE	Allows you to access resources to which the group is authorized
CREATE	Allows you to create RACF data set profiles for the group
CONNECT	Allows you to connect other users to the group
JOIN	Allows you to add new subgroups or users to the group, as well as assign group authorities to the new members

For more information on group authority, refer to *z/OS Security Server RACF Security Administrator's Guide*.

Access authority for data sets

Data sets can have one of the following access authorities:

NONE	Does not allow users to access the data set.
EXECUTE	For a private load library, EXECUTE allows users to load and execute, but not to read or copy, programs (load modules) in the library.

In order to specify EXECUTE for a private load library, you must ask for assistance from your RACF security administrator.

Attention

Anyone who has READ, UPDATE, CONTROL, or ALTER authority to a protected data set can create a copy of it. As owner of the copied data set, that user has control of the security characteristics of the copied data set, and can change them. For this reason, you should assign a UACC of NONE, and then selectively permit a small number of users to access your data set, as their needs become known. (See *z/OS Security Server RACF General User's Guide* for information on how to permit selected users or groups to access a data set.)

READ	Allows users to access the data set for reading only. (Note that users who can read the data set can copy or print it.)
UPDATE	<p>Allows users to read from, copy from, or write to the data set. UPDATE does not, however, authorize a user to delete, rename, move, or scratch the data set.</p> <p>Allows users to perform normal VSAM I/O (not improved control interval processing) to VSAM data sets.</p>
CONTROL	<p>For VSAM data sets, CONTROL is equivalent to the VSAM CONTROL password; that is, it allows users to perform improved control interval processing — CONTROL is control-interval access (access to individual VSAM data blocks), and the ability to retrieve, update, insert, or delete records in the specified data set.</p> <p>For non-VSAM data sets, CONTROL is equivalent to UPDATE.</p>
ALTER	ALTER allows users to read, update, delete, rename, move, or scratch the data set.

Basic information

When specified in a discrete profile, ALTER allows users to read, alter, and delete the profile itself *including the access list*.

ALTER does not allow users to change the owner of the profile using the ALTDSD command. However, if a user with ALTER access authority to a discrete data set profile renames the data set, changing the high-level qualifier to his or her own user ID, both the data set and the profile are renamed, *and* the OWNER of the profile is changed to the new user ID.

ALTER authority to a generic profile allows users to create new data sets that are covered by that profile, it does not give users authority over the profile itself.

Chapter 3. RACF TSO commands

Most RACF commands can be entered as RACF TSO commands. For a complete list of the RACF commands that can be entered as RACF TSO commands, see Table 2 on page 8.

RACF list commands, such as LISTUSER * or LISTGRP *, can generate many thousands of lines of output. This quantity of output is not very usable except as input to a processing program and can exhaust address space resources below the 16M line.

RACF commands that provide output listings (LISTDSD, LISTUSER, LISTGROUP, RLIST) are designed to be issued by users, not by programs. IBM does not support the processing of these commands by programs. The output format of these commands is not an intended interface and may change with any z/OS release or as the result of service (PTFs) applied within a release. Programs should not examine the output of these commands. Programs should use documented programming interfaces, such as: the output file from IRRDBU00 (database unload, which was designed specifically for this use), the results returned by RACROUTE REQUEST=EXTRACT, or the results returned by ICHEINTY.

The syntax of RACF TSO commands is the same as the syntax of TSO commands. For example, a comma or one or more blanks are valid delimiters for use between operands.

Notes:

1. "Syntax of RACF commands and operands" on page 9 contains the key to symbols used in the command syntax diagrams.
2. The TSO parse routines allow you to abbreviate an operand on a TSO command to the least number of characters that uniquely identify the operand. To avoid conflicts in abbreviations, it is a good practice to fully spell out all operands on commands that are hard-coded (as in programs and CLISTs, for example).
3. If you specify a keyword in the RACF segment multiple times on the same command, RACF uses only the last occurrence. If a keyword in a non-RACF segment (such as TSO, CICS, SESSION) is specified multiple times on the same command, the last occurrence is also used except for keywords where a list of values is valid (such as ADDUSER USER1 NETVIEW(OPCLASS(*value1 value2 value3 ...*))). For these keywords, all values on all specifications are accepted.
4. If you are sharing your RACF database with a VM system, you can administer VM classes from the MVS system. For RACF-provided VM classes see Supplied resource classes for z/VM and VM systems. Detailed information about working with VM classes can be found in the for RACF 1.10 information.
5. Make sure your job or logon specifies a large enough region size to run the commands, or they may ABEND unpredictably.

How to enter RACF TSO commands

The following sections describe how to enter RACF TSO commands. You can enter RACF TSO commands directly in the foreground during a TSO terminal session or by using the RACF ISPF panels.

TSO commands

You can enter commands in the background by using a batch job. You cannot use alias data set names in the RACF commands or panels. Alias names are alternative names for a data set that are defined in the catalog. RACF does not allow alias names because RACF uses the RACF database, not the catalog, for its processing.

Choosing between Using RACF TSO Commands and ISPF Panels

In general, you can perform the same RACF functions using RACF TSO commands and ISPF panels.

The **RACF TSO commands** provide the following advantages:

- Entering commands can be faster than displaying many panels in sequence.
- Using commands from book descriptions should be relatively straightforward. The examples in the books are generally command examples.
- Getting online help for RACF TSO commands

You can get online help for the RACF TSO commands documented in *z/OS Security Server RACF Command Language Reference*.

- To see online help for the PERMIT command, for example, enter:

```
HELP PERMIT
```

- To limit the information displayed, specify operands on the HELP command. For example, to see only the syntax of the PERMIT command, enter:

```
HELP PERMIT SYNTAX
```

Note: TSO online help is not available when RACF commands are entered as RACF operator commands.

- Getting message ID information

If a RACF TSO command fails, you will receive a message. If you do not get a message ID, enter:

```
PROFILE MSGID
```

Reenter the RACF TSO command that failed. The message appears with the message ID. See the *z/OS Security Server RACF Messages and Codes* for help if the message ID starts with ICH or IRR.

Note: PROFILE MSGID cannot be entered as a RACF operator command.

The **ISPF panels** provide the following advantages:

- When you use the panels, you avoid having to memorize a command and type it correctly. Panels can be especially useful if the command is complex or you perform a task infrequently.
- ISPF creates in the ISPF log a summary record of the work that you do. Unless you use the TSO session manager, the RACF commands do not create such a record.
- From the panels, you can press the HELP key to display brief descriptions of the fields on the panels.
- The options chosen when installing the RACF panels determine whether output (for example, profile listings, search results, and RACF options) is displayed in a scrollable form.
- The ISPF panels for working with password rules allow you to enter all of the password rules on one panel. Figure 2 on page 17 shows one of these panels.

```

RACF - SET PASSWORD FORMAT RULES
COMMAND ===>
Enter PASSWORD FORMAT RULES:

```

	MINIMUM LENGTH	MAXIMUM LENGTH	FORMAT
RULE 1:	—	—	_____
RULE 2:	—	—	_____
RULE 3:	—	—	_____
RULE 4:	—	—	_____
RULE 5:	—	—	_____
RULE 6:	—	—	_____
RULE 7:	—	—	_____
RULE 8:	—	—	_____

To cancel an existing rule, enter NO for MINIMUM LENGTH.
 To specify FORMAT, use the following codes for each character position:
 * = Any character A = Alphabetic C = Consonant V = Vowel
 W = No Vowel N = Numeric L = Alphanumeric

Figure 2. Sample ISPF Panel for RACF

Entering RACF TSO commands in the foreground

To enter RACF TSO commands in the foreground, you must be able to:

- Conduct a TSO terminal session
- Use the TSO commands
- Use system-provided aids (HELP command, attention interrupt, conversational messages)
- Respond to TSO prompts

See *z/OS TSO/E Primer* and *z/OS TSO/E Command Reference* for any information you need.

In addition, to enter RACF TSO commands from the foreground, you must be defined to the system.

The TSO LOGON command is used to define you to the system as a RACF user through the *user identity* (user ID), *password*, GROUP, and OIDCARD operands. To change your RACF password, you can use the *newpassword* operand on the LOGON command. If you have more than one account number defined in your TSO profile, you must supply an account number on the LOGON command.

The default data set name prefix in your TSO profile is used as the high-level qualifier of a DASD or tape data set name if you do not enter the fully-qualified name in a TSO or RACF command. RACF also uses the TSO default prefix as the high-level qualifier for the name of a CLIST created as a result of the RACF SEARCH command. If you do not have a prefix specified in your TSO profile, (PROFILE NOPREFIX), the userid from the SEARCH command issuer's ACEE is used as the qualifying prefix.

If you frequently use RACF TSO commands on RACF-protected data sets, you can set your TSO default prefix as follows:

- Set the default prefix to your user ID if you do a good deal of work with your own data sets.
- Set the default prefix to a specific RACF group name if you are working mostly with data sets from that group.

TSO commands

Remember that the command examples in this book use uppercase letters; but, when you are entering commands from a terminal, you can use either uppercase or lowercase letters.

Entering RACF TSO commands in the background

You can enter RACF TSO commands in the background by submitting a batch job as follows:

- Using the batch internal or remote reader facility of the job entry subsystem (JES)
- Using the TSO SUBMIT command from a terminal

The RACF data you need to enter on your JCL depends on whether the job entry subsystem (JES) at your installation includes the JES RACF user identification feature. If your level of JES includes the RACF user identification propagation feature, any jobs you submit to the background while logged onto TSO are automatically identified to RACF with the same user identifier. When the job runs, RACF uses your default group as your current connect group. (User, password, and group information is not required on the JOB statement, but if you do specify this information, it overrides the propagated specifications.)

If your level of JES does not include the RACF user identification propagation feature, you must include the USER, PASSWORD, and, optionally, GROUP parameters on the JCL JOB statement.

The USER, PASSWORD, and GROUP parameters on the JCL JOB statement identify you to the system as a RACF user. To change your RACF password, you can use the *new-password* operand of the PASSWORD command. For information on how to code these parameters, see *z/OS MVS JCL Reference*.

As an alternative to coding PASSWORD on JCL statements, you can use the TSO SUBMIT command (for systems that do not have the JES RACF user identification propagation feature) to automatically include this information during job submission. To use SUBMIT, you should code the USER (*userid*) and PASSWORD operands on the SUBMIT command. These operands are then put on the JCL JOB statement that the command generates. When the job runs, RACF uses the name of the user's default group as the current connect group.

Example of RACF TSO commands in the background

The following example shows how to submit RACF TSO commands in the background as a batch job:

```
//jobname      JOB      ...
//STEP1       EXEC    PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT    DD      SYSOUT=A
//SYSTSIN     DD      *
ADDGROUP      PROJECTA
ADDUSER       (PAJ5 ESH25)
ADDSD 'PROJECTA.XYZ.DATA'
PERMIT 'PROJECTA.XYZ.DATA' ID(PAJ5) ACCESS(UPDATE)
/*
```

When a fully-qualified data set name is not given in a command entered in the background, the effect is the same as for a command entered in the foreground; the user's TSO default data set name prefix is used as the high-level qualifier of a DASD or tape data set name. The prefix is also used as the high-level qualifier for the name of a command procedure (CLIST) created as a result of the RACF SEARCH command. If the user is defined to TSO, the default prefix is in the TSO

profile for the user specified in the USER parameter on the JCL JOB statement or the USER operand in the TSO SUBMIT command. If the user is not defined to TSO, there is no default prefix unless the TSO PROFILE command is used.

Chapter 4. RACF operator commands

The RACF operator commands allow you to perform RACF functions from an operator console. For a complete list of the RACF commands that can be entered as RACF operator commands, see Table 2 on page 8.

Note: Use of these commands requires that the RACF subsystem has been started. If the RACF subsystem has not been started at your installation, contact your system programmer.

These commands allow an MVS operator to perform certain RACF operations in the RACF subsystem. The RACF subsystem prefix in front of the command identifies the RACF subsystem as the processing environment. Some things to remember:

- RACF operator commands require an MVS/ESA™ environment with the RACF subsystem active.
- The DISPLAY and SIGNOFF commands require APPC/MVS and persistent verification.
- If a command can be issued as both a RACF TSO command or a RACF operator command:
 - RACF first checks that the issuer is defined to RACF and if not, an error message is issued.

If you are defined to RACF, RACF verifies that you have sufficient authority to the proper resource in the OPERCMDS class to determine if you have authority to issue the command as an operator command. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

If the OPERCMDS class is not active, or if no OPERCMDS profile exists, then the user is allowed to issue the command as a RACF operator command.
 - You must be logged on to the console to issue the command as a RACF operator command.

Note: The RVARY command is the exception, because it can be issued as both a RACF TSO command and a RACF operator command but does not have to fit these circumstances.

- If a command can be issued as a RACF operator command, but not as a RACF TSO command:
 - RACF first checks to see if you have OPERCMDS authority. If the user is logged on to the console, the OPERCMDS class is active, and a OPERCMDS profile exists, you have OPERCMDS authority.
 - If you are not logged on to the console, the OPERCMDS class is inactive, or no OPERCMDS profile exists, you can only issue the command from the master console or a console with system authority.

Note: The DISPLAY command is the exception, because it can be issued under these circumstances without any particular console authority.

Rules for entering RACF operator commands

1. A RACF operator command must contain the RACF subsystem prefix. A command such as the DISPLAY command could be entered on the command line as follows:

#DISPLAY xxxx

Operator commands

where:

= subsystem prefix. The subsystem prefix specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your system programmer.

If no subsystem prefix has been defined, and the subsystem name is rac1, the same command would be entered as follows:

```
rac1 DISPLAY xxxx
```

Note: If you need to find out what the subsystem prefix is, contact your system programmer.

xxxx = DISPLAY operands

2. Separate operands with commas. Do not specify commas before the first operand or after the last operand.

For example, enter a DISPLAY command with two operands as follows:

```
#DISPLAY xxxx,yyyy
```

3. You can also separate operands with blanks. This practice is not encouraged, however, because future releases might not allow this.
4. The order in which you specify the operands on the command line does not affect the command.

For example:

```
#DISPLAY xxxx,yyyy
```

and

```
#DISPLAY yyyy,xxxx
```

gives the same result.

5. RACF commands entered as RACF operator commands must meet the MVS restrictions on command length and operand content.

Chapter 5. RACF Command Syntax

This topic describes the syntax and function of the RACF commands. The commands are presented in alphabetical order. Each command descriptions contains several examples.

ADDGROUP (Add Group Profile)

Purpose

Use the ADDGROUP command to define a new group to RACF.

The command adds a profile for the new group to the RACF database. It also establishes the relationship of the new group to the superior group you specify.

Group profiles consist of a RACF segment and, optionally, other segments such as DFP and OMVS. You can use this command to specify information in any segment of the profile.

Issuing Options

The following table identifies the eligible options for issuing the ADDGROUP command:

Table 3. How the ADDGROUP Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	Yes	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To delete a group profile, see “DELGROUP (Delete Group Profile)” on page 186.
- To change a group profile, see “ALTGROUP (Alter Group Profile)” on page 104.
- To connect a user to a group, see “CONNECT (Connect User to Group)” on page 173.
- To remove a user from a group, see “REMOVE (Remove User from Group)” on page 376.
- To obtain a list of group profiles, see “SEARCH (Search RACF Database)” on page 408.
- To list a group profile, see “LISTGRP (List Group Profile)” on page 214.

Authorization Required

To use the ADDGROUP command, you must meet at least one of the following conditions:

- Have the SPECIAL attribute,
- Have the group-SPECIAL attribute and the superior group is within your group-SPECIAL scope,
- Be the owner of the superior group, or
- Have JOIN authority in the superior group.

When issuing the ADDGROUP command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

To add DFP, OMVS, or OVM suboperands to a group's profile you must meet at least one of the following conditions:

- You must have the SPECIAL attribute.
- Your installation must permit you to do so through field-level access checking.

For information on field-level access checking, see *z/OS Security Server RACF Security Administrator's Guide*.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

To specify the SHARED keyword, you must have the SPECIAL attribute or at least READ authority to the SHARED.IDS resource in the UNIXPRIV class.

Syntax

```
[subsystem-prefix]{ADDGROUP | AG}

    (group-name ...)
    [ AT([node].userid ...) | ONLYAT([node].userid ...)]
    [ DATA('installation-defined-data') ]
    [ DFP(
        [ DATAAPPL(application-name) ]
        [ DATACLAS(data-class-name) ]
        [ MGMTCLAS(management-class-name) ]
        [ STORCLAS(storage-class-name) ] ) ]
    [ MODEL(dsname) ]
    [ OMVS
        [ (
            AUTOGID
            | GID(group-identifier) [ SHARED ]
        ) ] ]
    [ OVM [ ( GID(group-identifier) ) ] ]
    [ OWNER(userid or group-name) ]
    [ SUPGROUP(group-name) ]
    [ TERMUACC | NOTERMUACC ]
    [ TME(
        [ ROLES(profile-name ...) ] ) ]
    [UNIVERSAL]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

ADDGROUP

Parameters

subsystem-prefix

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

group-name

Specifies the name of the group whose profile is to be added to the RACF database. If you are defining more than one group, the list of group names must be enclosed in parentheses.

This operand is required and must be the first operand following ADDGROUP. Each *group-name* must be unique and must not currently exist in the RACF database as a group name or a user ID.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([*node*].*userid* ...)

Specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT([*node*].*userid* ...)

Specifies that the command is to be directed only to the node specified by *node* where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed only to the local node.

DATA('installation-defined-data')

Specifies up to 255 characters of installation-defined data to be stored in the group profile and must be enclosed in quotes. It might also contain double-byte character set (DBCS) data.

Use the LISTGRP command to list this information.

DFP

Specifies that when you define a group to RACF, you can enter any of the following suboperands to specify default values for the DFP data, management, and storage classes. DFP uses this information to determine data management and DASD storage characteristics when a user creates a new group data set.

DATAAPPL(*application-name*)

Specifies DFP data application identifier. The maximum length of a data class name is 8 characters.

DATACLAS(*data-class-name*)

Specifies the default data class. The maximum length of a data class name is 8 characters.

A data class can specify some or all of the physical data set attributes associated with a new data set. During new data set allocation, data management uses the value you specify as a default unless it is preempted by a higher priority default, or overridden in some other way (for example, by JCL).

Note: The value you specify must be a valid data class name defined for use on your system. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

For information on defining DFP data classes, see *z/OS DFSMSdfp Storage Administration Reference*.

MGMTCLAS(*management-class-name*)

Specifies the default management class. The maximum length of a management class name is 8 characters.

A management class contains a collection of management policies that apply to data sets. Data management uses the value you specify as a default unless it is preempted by a higher priority default, or overridden in some other way (for example, by JCL).

Note: The value you specify must be defined as a profile in the MGMTCLAS general resource class, and the group must be granted at least READ access to the profile. Otherwise, RACF does not allow the group access to the specified MGMTCLAS. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

For information on defining DFP management classes, see *z/OS DFSMSdfp Storage Administration Reference*.

STORCLAS(*storage-class-name*)

Specifies the default storage class. The maximum length of a *storage-class-name* is 8 characters.

A storage class specifies the service level (performance and availability) for data sets managed by the Storage Management Subsystem (SMS). During new data set allocation, data management uses the value you specify as a default unless it is preempted by a higher priority default, or overridden in some other way (for example, by JCL).

Note: The value you specify must be defined as a profile in the STORCLAS general resource class, and the group must be granted at least READ access to the profile. Otherwise, RACF does not allow the group access to the specified STORCLAS. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

For information on defining DFP storage classes, see *z/OS DFSMSdfp Storage Administration Reference*.

MODEL(*dsname*)

Specifies the name of a discrete MVS data set profile to be used as a model for new *group-name* data sets. For this operand to be effective, the MODEL(GROUP) option (specified on the SETROPTS command) must be active.

RACF always prefixes the data set name with *group-name* when it accesses the model.

ADDGROUP

For information about automatic profile modeling, refer to *z/OS Security Server RACF Security Administrator's Guide*.

OMVS

Specifies z/OS UNIX System Services information for the group being defined to RACF.

AUTOGID | GID

Specifies whether RACF is to automatically assign an unused GID value to the group or whether a specific GID value is to be assigned.

AUTOGID

Specifies that RACF is to automatically assign an unused GID value to the group. The GID value is derived from information obtained from the BPX.NEXT.USER profile in the FACILITY class. For more information on setting up BPX.NEXT.USER, see *z/OS Security Server RACF Security Administrator's Guide*.

If you are using RRSF automatic command direction for the GROUP class, the command sent to other nodes will contain an explicit assignment of the GID value which was derived by RACF on the local node.

Rules:

- AUTOGID cannot be specified if more than one group is entered.
- The AUTOGID keyword is mutually exclusive with the SHARED keyword.
- If both GID and AUTOGID are specified, AUTOGID is ignored.
- Field- level access checking for the GID field applies when using AUTOGID.

GID(group-identifier) [SHARED]

GID(group-identifier)

Specifies the group identifier. The GID is a numeric value between 0 and 2 147 483 647.

When a GID is assigned to a group, all users connected to that group who have a user identifier (UID) in their user profile can use functions such as the TSO/E command, OMVS, and can access the hierarchical file system (HFS) based on the GID and UID values assigned.

Notes:

1. If the security administrator has defined the SHARED.IDS profile in the UNIXPRIV class, the GID must be unique. Use the SHARED keyword in addition to GID to assign a value that is already in use.
2. If SHARED.IDS is not defined, RACF does not require the GID to be unique. The same value can be assigned to multiple groups, but this is not recommended because individual group control would be lost. However, if you want a set of groups to have exactly the same access to z/OS UNIX resources, you might decide to assign the same GID to more than one group.
3. RACF allows you to define and connect a user to more than 300 groups (which is the same as the NGROUPS_MAX variable defined in the POSIX standard), but when a process is created or z/OS UNIX group information is requested, only up to the first 300 z/OS UNIX groups are associated with the process or user.

ADDGROUP

The first 300 z/OS UNIX groups, that have GIDs, to which a user is connected are used by z/OS UNIX . LISTUSER displays the groups in the order that RACF examines them when determining which of the user's groups are z/OS UNIX groups. See *z/OS UNIX System Services Planning* for information on NGROUPS_MAX.

SHARED

If the security administrator has chosen to control the use of shared GIDs, this keyword must be used in addition to the GID keyword to specify the group identifier if it is already in use by at least one other group. The administrator controls shared GIDs by defining the SHARED.IDS profile in the UNIXPRIV class.

Rules:

- If the SHARED.IDS profile is not defined, SHARED is ignored.
- If SHARED is specified in the absence of GID, it is ignored.
- If the SHARED.IDS profile is defined and SHARED is specified, but the value specified with GID is not currently in use, SHARED is ignored and UNIXPRIV authority is not required.
- Field - level access checking for the GID field applies when using SHARED.
- The SHARED keyword is mutually exclusive with the AUTOGID keyword.

OVM

Specifies OpenExtensions VM information for the group being defined to RACF.

GID(*group-identifier*)

specifies the OpenExtensions VM group identifier. The GID is a numeric value between 0 and 2 147 483 647.

If you do not specify GID, the group is assigned the default GID of 4 294 967 295 (X'FFFFFFFF') and a LISTGRP shows NONE for the GID.

Notes:

1. RACF does not require the GID to be unique. The same value can be assigned to multiple groups, but this is not recommended because individual group control would be lost. However, if you want a set of groups to have exactly the same access to the OpenExtensions VM resources, you might decide to assign the same GID to more than one group.
2. The value defined for the NGROUPS_MAX variable in the ICHNGMAX macro on VM defines the maximum number of OpenExtensions VM groups to be associated with an OpenExtensions VM process or user. The NGROUPS_MAX variable on VM is a number between 32 and 125, inclusive. However, RACF allows you to define and connect a user to more than the number of groups defined in this variable. If the NGROUPS_MAX variable is *n* and a process is created or OpenExtensions VM group information is requested, only up to the first *n* OpenExtensions VM groups are associated with the process or user. The first *n* OpenExtensions VM groups to which a user is connected are used by OpenExtensions VM. LISTUSER displays the groups in the

ADDGROUP

order that RACF examines them when determining which of the user's groups are OpenExtensions VM groups.

See *z/OS Security Server RACF Macros and Interfaces* for information on NGROUPS_MAX.

OWNER(*userid or group-name*)

Specifies a RACF-defined user or group to be assigned as the owner of the new group. If you do not specify an owner, you are defined as the owner of the group. If you specify a group name, it must be the name of the superior group for the group you are adding.

SUPGROUP(*group-name*)

Specifies the name of an existing RACF-defined group. This group becomes the superior group of the group profile you are defining.

If you omit SUPGROUP, RACF uses your current connect group as the superior group.

If you specify a group name and also specify OWNER with a group name, you must use the same group name on both SUPGROUP and OWNER.

If your authority to issue ADDGROUP comes from the group-SPECIAL attribute, any group you specify must be within the scope of the group in which you are a group-SPECIAL user.

TERMUACC | NOTERMUACC

TERMUACC

Specifies that during terminal authorization checking, RACF allows any user in the group access to a terminal based on the universal access authority for that terminal. TERMUACC is the default value if you omit both TERMUACC and NOTERMUACC.

NOTERMUACC

Specifies that the group or a user connected to the group must be explicitly authorized (through the PERMIT command with at least READ authority) to access a terminal.

TME

Specifies that information for the Tivoli Security Management Application is to be added.

Note: The TME segment fields are intended to be updated only by the Tivoli Security Management Application, which manages updates, permissions, and cross references. A security administrator should only directly update Tivoli Security Management fields on an exception basis.

ROLES(*profile-name ...*)

Specifies a list of roles that reference this group.

Profile-name should be the name of a defined role, which is a discrete general resource profile in the ROLE class.

UNIVERSAL

Specifies that this is a universal group that allows an effectively unlimited number of users to be connected to it for the purpose of resource access. The number of users in a universal group with authority higher than USE, or with the attributes SPECIAL, OPERATIONS or AUDITOR at the group level, is still limited to 5957.

When displayed with the LISTGRP command, not all group members will be listed. Only users with authority higher than USE or with the attributes SPECIAL, OPERATIONS or AUDITOR at the group level will be shown in the member list.

Examples

Table 4. ADDGROUP Examples

Example 1	<i>Operation</i>	User IA0 wants to add the group PROJECTA as a subgroup of RESEARCH. User IA0 will be the owner of group PROJECTA. Users in group PROJECTA will be allowed to access a terminal based on the universal access authority assigned to that terminal.
	<i>Known</i>	User IA0 has JOIN authority to group RESEARCH. User IA0 is currently connected to group RESEARCH. User IA0 wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.
	<i>Command</i>	@ADDGROUP PROJECTA
Example 2	<i>Defaults</i>	SUPGROUP(RESEARCH) OWNER(IA0) TERMUACC
	<i>Operation</i>	User ADM1 wants to add the group PROJECTB as a subgroup of RESEARCH. Group RESEARCH will be the owner of group PROJECTB. Group PROJECTB must be authorized to use terminals through the PERMIT command.
	<i>Known</i>	User ADM1 has JOIN authority to group RESEARCH. User ADM1 is currently connected to group SYS1. USER ADM1 wants to issue the command as a RACF TSO command.
Example 3	<i>Command</i>	ADDGROUP PROJECTB SUPGROUP(RESEARCH) OWNER(RESEARCH) NOTERMUACC
	<i>Defaults</i>	None
	<i>Operation</i>	User ADM1 wants to add the group SYSINV as a subgroup of RESEARCH. This group will be used as the administrative group for RACF and will use a model name of 'SYSINV.RACF.MODEL.PROFILE'. User ADM1 wants to direct the command to run under the authority of user APW02.
	<i>Known</i>	User APW02 has JOIN authority to group RESEARCH and ADM1 wants to issue the command as a RACF TSO command.
		ADM1 and APW02 have an established user ID association.
	<i>Command</i>	ADDGROUP SYSINV SUPGROUP(RESEARCH) MODEL(RACF.MODEL.PROFILE) DATA('RACF ADMINISTRATION GROUP') AT(.APW02)
	<i>Defaults</i>	OWNER(ADM1) TERMUACC
		Command direction defaults to the local node.

ADDGROUP

Table 4. ADDGROUP Examples (continued)

Example 4

Operation User ADM1 wants to add the group DFPADMN as a subgroup of SYSADMN. Group SYSADMN will be the owner of group DFPADMN. Users in group DFPADMN will be allowed to access a terminal based on the universal access authority assigned to that terminal. Group DFPADMN will be assigned the following default information to be used for new DFP-managed data sets created for the group:

- Data class DFP2DATA
- Management class DFP2MGMT
- Storage class DFP2STOR
- Data application identifier DFP2APPL.

- Known**
- User ADM1 has JOIN authority to group SYSADMN.
 - User ADM1 is currently connected to group SYS1.
 - User ADM1 has field-level access of ALTER to the fields in the DFP segment.
 - DFP2MGMT has been defined to RACF as a profile in the MGMTCLAS general resource class, and group DFPADMN has been given READ access to this profile.
 - DFP2STOR has been defined to RACF as a profile in the STORCLAS general resource class, and group DFPADMN has been given READ access to this profile.
 - User ADM1 wants to issue the command as a RACF TSO command.

Command ADDGROUP DFPADMN SUPGROUP(SYSADMN) OWNER(SYSADMN)
DFP(DATACLAS(DFP2DATA) MGMTCLAS(DFP2MGMT) STORCLAS(DFP2STOR)
DATAAPPL(DFP2APPL))

Defaults TERMUACC

Example 5

Operation User ADM1 wants to add the UNIVERSAL group NETGROUP as a subgroup of SYS1. User IBMUSER will be the owner of group NETGROUP. Universal group NETGROUP can have an unlimited number of members (that have USE authority and are not SPECIAL, OPERATIONS, or AUDITOR).

- Known**
- User ADM1 has group-SPECIAL authority to group SYS1.
 - User ADM1 is currently connected to group SYS1.

Command ADDGROUP NETGROUP DATA('INTERNET CUSTOMER GROUP') SUPGROUP(SYS1)
OWNER(IBMUSER) UNIVERSAL

Defaults None apply

Example 6

Operation User RACFADM with SPECIAL or UPDATE authority requests the addition of a new z/OS UNIX group. The user specifies AUTOGID so that RACF will automatically assign an unused GID to the new user.

Known The group profile is owned by RACFADM and belongs to RACFADM's current connect group SYSOM. The BPX.NEXT.USER profile in the FACILITY class has been set up to allow automatic GID assignment.

Command ADDGROUP UNIXGRP OMVS(AUTOGID HOME('/u/unixgrp') CPUTIMEMAX(5000)
ASSIZEMAX(40000000))

Defaults DFLTGRP(SYSOM) OWNER(RACFADM)

ADDSD (Add Data Set Profile)

Purpose

Use the ADDSD command to add RACF protection to data sets with either discrete or generic profiles.

Changes made to discrete profiles take effect after the ADDSD command is processed. Changes made to generic profiles do not take effect until one or more of the following steps is taken:

- The user of the data set issues the LISTDSD command:

```
LISTDSD DA(data-set-protected-by-the-profile) GENERIC
```

Note: Use the data set name, not the profile name.

- The security administrator issues the SETROPTS command:

```
SETROPTS GENERIC(DATASET) REFRESH
```

See SETROPTS command for authorization requirements.

- The user of the data set logs off and logs on again.

For more information, refer to *z/OS Security Server RACF Security Administrator's Guide*.

Issuing Options

The following table identifies the eligible options for issuing the ADDSD command:

Table 5. How the ADDSD Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	Yes	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To change a data set profile, see “ALTDSD (Alter Data Set Profile)” on page 90.
- To delete a data set profile, see “DELDSD (Delete Data Set Profile)” on page 181.
- To permit or deny access to a data set profile, see “PERMIT (Maintain Resource Access Lists)” on page 247.
- To obtain a list of data set profiles, see “SEARCH (Search RACF Database)” on page 408.
- To list a data set profile, see “LISTDSD (List Data Set Profile)” on page 200.

ADDSD

Authorization Required

When issuing the ADDSD command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

Notes:

1. You need not have the SPECIAL attribute to specify the OWNER operand.
2. To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).
3. To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

The level of authority you need to use the ADDSD command and the types of profiles you can define are:

- To protect a user data set with RACF, one of the following must be true:
 - The high-level qualifier of the data set name (or the qualifier supplied by the RACF naming conventions table or by a command installation exit) must match your user ID.
 - You must have the SPECIAL attribute.
 - The user ID for the data set profile must be within the scope of a group in which you have the group-SPECIAL attribute.
- To protect a group data set with RACF, one of the following must be true:
 - You must have at least CREATE authority in the group.
 - You must have the SPECIAL attribute.
 - You must have the OPERATIONS attribute and not be connected to the group.
 - The data set profile must be within the scope of the group in which you have the group-SPECIAL attribute.
 - The data set profile must be within the scope of the group in which you have the group-OPERATIONS attribute, and you must not be connected to the group.

Notes:

1. If you have the OPERATIONS or group-OPERATIONS attribute and are connected to a group, you must have at least CREATE authority in that group to protect a group data set.
 2. When creating a group data set profile, the profile creator's user ID is placed on the access list with ALTER authority unless the creation was allowed due to OPERATIONS or group-OPERATIONS authority or unless the SETROPTS NOADDCREATOR option is in effect.
 3. To protect a group data set where the high-level qualifier of the data set name is VSAMDSET, you need neither CREATE authority in the VSAMDSET group nor the SPECIAL attribute. (A universal group authority of CREATE applies to the RACF-defined VSAMDSET group.)
- To define to RACF a data set that was brought from another system where it was RACF-indicated and RACF-protected with a discrete profile, one of the following must be true:
 - You must either have the SPECIAL attribute, or the data set's profile is within the scope of a group in which you have the group-SPECIAL attribute

- Your user ID must be the high-level qualifier of the data set name (or the qualifier supplied by the naming conventions routine or a command installation exit).
- To assign a security category to a profile, you must have the SPECIAL attribute or have the category in your user profile.
- To assign a security level to a profile, you must have the SPECIAL attribute or, in your own profile, a security level that is equal to or greater than the security level you are defining.
- To assign a security label to a profile, you must have the SPECIAL attribute or READ authority to the security label profile. However, the security administrator can limit the ability to assign security labels to only users with the SPECIAL attribute.
- To access the DFP or TME segment, field-level access checking is required.
- When either a user or group uses modeling to protect a data set with a discrete profile, RACF copies the following fields from the model profile: the level number, audit flags, global audit flags, the universal access authority (UACC), the owner, the warning, the access list, installation data, security category names, the security level name, the user to be notified, the retention period for a tape data set, and the erase indicator.
- To add a discrete profile for a VSAM data set already RACF-protected by a generic profile, you must have ALTER access authority to the catalog or to the data set through the generic profile.

Model Profiles

To specify a model data set profile (using, as required, FROM, FCLASS, FGNERIC, and FVOLUME), you must have sufficient authority over the model profile — the “from” profile. RACF makes the following checks until one of the conditions is met:

- You have the SPECIAL attribute.
- The “from” profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You are the owner of the “from” profile.
- The high-level qualifier of the profile name (or the qualifier supplied by the naming conventions routine or a command installation exit routine) is your user ID.

For discrete profiles only:

- You are on the access list in the “from” profile with ALTER authority. (If you have any lower level of authority, you cannot use the profile as a model.)
- Your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is on the access list in the “from” profile with ALTER authority. (If any group that RACF checked has any lower level of authority, you cannot use the profile as a model.)
- The UACC is ALTER.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the ADDSD command is:

[subsystem-prefix]{ADDSD | AD}

ADDSD

```
(profile-name-1 [/password] ...)
[ ADDCATEGORY(category-name ...) ]
[ AT([node].userid ...) | ONLYAT([node].userid ...) ]
[ AUDIT( access-attempt [(audit-access-level)] ...) ]
[ DATA('installation-defined-data') ]
[ DFP(RESOWNER(userid or group-name) | NORESOWNER) ]
[ ERASE ]
[ FCLASS(profile-name-2-class) ]
[ FGENERIC ]
[ FILESEQ(number) ]
[ FROM(profile-name-2) ]
[ FVOLUME(profile-name-2-serial) ]
[ {GENERIC | MODEL | TAPE} ]
[ LEVEL(nn) ]
[ {NOSET | SET | SETONLY} ]
[ NOTIFY [(userid) ] ]
[ OWNER(userid or group-name) ]
[ RETPD(nnnnn) ]
[ SECLABEL(security-label) ]
[ SECLEVEL(security-level) ]
[ TME( [ ROLES(role-access-specification ...) ] ) ]
[ UACC(access-authority) ]
[ UNIT(type) ]
[ VOLUME(volume-serial ...) ]
[ WARNING ]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

profile-name-1

specifies the name of the discrete or generic profile to be added to the RACF database. If you specify more than one name, the list of names must be enclosed in parentheses.

The format of the profile name should follow the TSO/E data set naming conventions, except that the high-level qualifier of the profile name (or the qualifier determined by the naming conventions table or by a command

installation exit) must be a user ID or a group name. See the *z/OS Security Server RACF Security Administrator's Guide* for more information about the TSO/E data set naming conventions.

To specify a user ID other than your own, you must have the SPECIAL attribute, or the data set profile must be within the scope of a group in which you have the group-SPECIAL attribute. To define a group data set where the high-level qualifier of the data set name is not VSAMDSSET, you must have at least CREATE authority in the specified group, or the SPECIAL attribute, or the data set must be within the scope of a group in which you have the group-SPECIAL attribute.

This operand is required and must be the first operand following ADDSD. Note that, because RACF uses the RACF database and not the system catalog, you cannot use alias data set names.

For additional information, see "Profile Names for Data Sets" on page 496 and the section describing rules for defining data set profiles in *z/OS Security Server RACF Security Administrator's Guide*.

Tape Data Set: If you are defining a discrete profile that protects a tape data set, you must specify TAPE. If you are defining more than one tape data set profile, the data sets must all reside on the same volume, and you must specify the profile names in an order that corresponds to the file sequence numbers of the data sets on the volume.

VSAM Data Set: All of the components of a VSAM data set are protected by the profile that protects the cluster name. It is not necessary to create profiles that protect the index and the data components of the cluster.

Data Sets Catalogued by an Indirect VOLSER: Data sets catalogued by an indirect VOLSER should be protected by a fully-qualified generic profile, or by an ADDSD with the real unit and volume for each data set covered by that catalog entry. The latter must be done while the data set is online.

/password

specifies the data set password if you are protecting an existing password-protected data set. If you specify a generic or model profile, RACF ignores this operand.

For a non-VSAM password-protected data set, the WRITE level password must be specified.

For a VSAM data set that is not password-protected, you do not need the password or RACF access authority for the catalog.

A password is not required when you specify NOSET.

If the command is executing in the foreground and you omit the password for a password-protected data set, the logon password is used. You are prompted if the password you enter or the logon password is incorrect. (If it is a non-VSAM multivolume data set, you are prompted once for each volume on which the data set resides.)

If the command is executing in a batch job and you either omit the password for a password-protected data set or supply an incorrect password, the operator is prompted. (If it is a non-VSAM multivolume data set, the operator is prompted once for each volume on which the data set resides.)

ADDCATEGORY(*category-name ...*)

specifies one or more names of installation-defined security categories. The

ADDSD

names you specify must be defined as members of the CATEGORY profile in SECDATA class. (For information on defining security categories, see *z/OS Security Server RACF Security Administrator's Guide*.)

When the SECDATA class is active and you specify ADDCATEGORY, RACF performs security category checking in addition to its other authorization checking. If a user requests access to a data set, RACF compares the list of security categories in the user's profile with the list of security categories in the data set profile. If RACF finds any security category in the data set profile that is not in the user's profile, RACF denies access to the data set. If the user's profile contains all the required security categories, RACF continues with other authorization checking.

Note: RACF does not perform security category checking for a started task or user that has the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF started procedures table or STARTED class, or to other users by customer-supplied RACF exits.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT(*[node].userid ...*)

specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT(*[node].userid ...*)

specifies that the command is to be directed only to the node specified by *node* where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed only to the local node.

AUDIT(*access-attempt*[(*audit-access-level*)...]]

access-attempt

specifies which access attempts you want to log on the SMF data set. The following options are available:

ALL

indicates that you want to log both authorized accesses and detected unauthorized access attempts.

FAILURES

indicates that you want to log detected unauthorized attempts.

NONE

indicates that you do not want any logging to be done.

SUCCESS

indicates that you want to log authorized accesses.

audit-access-level

specifies which access levels you want logged on the SMF data set. The levels you can specify are:

ALTER

logs ALTER access-level attempts only.

CONTROL

logs access attempts at the CONTROL and ALTER levels.

READ

logs access attempts at any level. READ is the default value if you omit audit-access-level.

UPDATE

logs access attempts at the UPDATE, CONTROL, and ALTER levels.

FAILURES(READ) is the default value if you omit the AUDIT operand. You cannot audit access attempts at the EXECUTE level.

DATA('installation-defined-data')

specifies up to 255 characters of installation-defined data to be stored in the data set profile and must be enclosed in quotes. It might also contain double-byte character set (DBCS) data.

Use the LISTDSD command to list this information.

DFP

specifies that for an SMS-managed data set, you can enter the following information:

RESOWNER(userid or group-name) | NORESOWNER

specifies the user ID or group name of the actual owner of the data sets protected by the profile specified in *profile-name-1*. This name must be that of a RACF-defined user or group. (The data set resource owner, specified with RESOWNER, is distinguished from the owner specified with OWNER, which represents the user or group that owns the data set profile).

If NORESOWNER is specified, the user or group represented by the high level qualifier of the data set profile is assigned as the owner of data sets protected by the profile when SMS needs to determine the RESOWNER.

ERASE

specifies that when SETROPTS ERASE(NOSECLEVEL) is active, data management is to physically erase the DASD data set extents at the time the data set is deleted (scratched) or released for reuse. Erasing the data set means overwriting all allocated extents with binary zeros.

This operand is ignored for the following:

- If the data set is not a DASD data set
- If SETROPTS ERASE(ALL) is specified for your installation (user and data set profile definitions are overridden)
- SETROPTS ERASE(SECLEVEL(*security_level*)) is specified for your installation (data sets equal or higher in security level are always erased, while those lower in security level are never erased)

FCLASS(profile-name-2-class)

specifies the name of the class to which *profile-name-2* belongs. The valid class names are DATASET and those classes defined in the class descriptor table. If you omit this operand, RACF assumes the DATASET class. This operand is valid only when you also specify the FROM operand; otherwise, RACF ignores it.

FGENERIC

specifies that RACF is to treat *profile-name-2* as a generic name, even if it is fully-qualified (meaning that it does not contain any generic characters). This operand is only needed when *profile-name-2* is a DATASET profile.

FILESEQ(*number*)

specifies the file sequence number for a tape data set. The number can range from 1 through 9999.

If you specify more than one *profile name*, RACF assigns the file sequence number that you specify to the first profile name, then increments the number by one for each additional name. Thus, be sure to specify profile names in the order of their file sequence numbers.

If you omit FILESEQ and specify VOLUME, the default is FILESEQ(1). If you omit both FILESEQ and VOLUME, RACF retrieves the file sequence number and volume serial number from the catalog.

If you omit TAPE, RACF ignores FILESEQ.

FROM(*profile-name-2*)

specifies the name of an existing discrete or generic profile that RACF is to use as a model for the new profile. The model profile name you specify on the FROM operand overrides any model name specified in your user or group profile. If you specify FROM and omit FCLASS, RACF assumes that *profile-name-2* is the name of a profile in the DATASET class.

To specify FROM, you must have sufficient authority to both *profile-name-1* and *profile-name-2*, as described in “Authorization Required” on page 34.

Naming conventions processing affects *profile-name-2* in the same way that it affects *profile-name-1*.

Mixed case profile names are accepted and preserved when FCLASS refers to a class defined in the class descriptor table with CASE=ASIS.

If the profile being added is for a group data set and the user has the GRPACC attribute for that group, RACF places the group on the access list with UPDATE access authority. Otherwise, if the group is already on the access list, RACF changes the group’s access authority to UPDATE.

Possible Changes to Copied Profiles When Modeling Occurs

When a profile is copied during profile modeling, the new profile could differ from the model in the following ways:

- RACF places the user on the access list with ALTER access authority or, if the user is already on the access list, changes the user’s access authority to ALTER. This does not occur if the NOADDCREATOR option is in effect.
If the profile being added is for a group data set and the user has the GRPACC attribute for that group, RACF places the group on the access list with UPDATE access authority. If the group is already on the access list, RACF changes the group’s access authority to UPDATE. These access list changes do not occur if the data set profile is created only because the user has the OPERATIONS attribute.
- The security label, if specified in the model profile, is not copied. Instead, the user’s current security label is used.
- Information in the non-RACF segments (for example, the DFP segment) is not copied.

FVOLUME(*profile-name-2-serial*)

specifies the volume RACF is to use to locate the model profile (*profile-name-2*).

If you specify FVOLUME and RACF does not find *profile-name-2* associated with that volume, the command fails. If you omit this operand and the data set name appears more than once in the RACF database, the command fails.

FVOLUME is valid only when FCLASS either specifies or defaults to DATASET and when *profile-name-2* specifies a discrete profile. Otherwise, RACF ignores FVOLUME.

GENERIC | MODEL | TAPE

GENERIC

specifies that RACF is to treat *profile-name-1* as a fully-qualified generic name, even if it does not contain any generic characters.

MODEL

specifies that you are defining a model profile to be used when new data sets are created. The SETROPTS command (specifying MODEL operand with either GROUP or USER) controls whether this profile is used for data sets with group names or user ID names.

When you specify MODEL, you can omit UNIT and VOLUME.

When you specify MODEL, the SET, GENERIC, and TAPE operands are ignored, and NOSET is used as the default.

MODEL and GENERIC operands are mutually exclusive. You cannot specify a generic profile for automatic profile modelling through the MODEL operand of ADDUSER, ALTUSER, ADDGROUP, or ALTGROUP. However, you can explicitly use a generic profile as a model with the FROM operand, and if needed, the FGENERIC operand of the ADDSD command.

For information about automatic profile modeling, refer to *z/OS Security Server RACF Security Administrator's Guide*.

TAPE

specifies that the data set profile is to protect a tape data set. If tape data set protection is not active, RACF treats TAPE as an invalid operand and issues an appropriate error message. If *profile-name-1* is a generic profile name, RACF ignores this operand. (RACF processes a tape data set protected by a generic profile in the same way as it processes a DASD data set protected by a generic profile.)

LEVEL(*nn*)

specifies a level indicator, where *nn* is an integer between 0 and 99. The default is 0.

Your installation assigns the meaning of the value.

RACF includes it in all records that log data set accesses and in the LISTDSD command display.

NOSET | SET | SETONLY

NOSET

specifies that the data set is not to be RACF-indicated.

For a DASD data set, use NOSET when you are defining a data set to RACF that has been brought from another system where it was RACF-protected. (The data set is already RACF-indicated.)

For a tape data set, use NOSET when, because of a previous error, the TVTOC indicates that the data set is RACF-indicated, but the discrete profile is missing.

If you specify NOSET, for a discrete profile, when the data set is not already RACF-indicated, RACF access control to that data set is not enforced.

ADDSD

If you specify NOSET, the volumes on which the data set or catalog resides need not be online, and the password in the first operand of this command is not required.

To use NOSET, one of the following must be true:

- You must have the SPECIAL attribute
- The profile must fall within the scope of a group in which you have the group-SPECIAL attribute
- The high-level qualifier of the data set name (or the qualifier supplied by a command installation exit routine) must be your user ID.

If you specify a generic profile name, RACF ignores this operand.

Note: If you specify a profile name that exists as a generation data group (GDG) data set base name with NOSET—but do not specify a unit and volume, RACF creates a model profile for the data set instead of a discrete profile. In this situation, the model profile provides the same protection as a discrete profile.

SET

specifies that the data set is to be RACF-indicated. SET is the default value when you are RACF-protecting a data set. If the indicator is already on, the command fails. If you specify a generic profile name or the GENERIC operand, RACF ignores this operand.

SETONLY

specifies that for a tape data set, RACF is to create only an entry in the TVTOC; it is not to create a discrete data set profile. Specifying SETONLY allows you to protect a tape data set with a TVTOC and a generic profile.

Thus, you would normally specify SETONLY with TAPE, and, when you do, RACF ignores the OWNER, UACC, AUDIT, DATA, WARNING, LEVEL, and RETPD operands. If you specify SETONLY without TAPE, RACF treats SETONLY as SET.

NOTIFY[(userid)]

specifies the user ID of a RACF-defined user to be notified whenever RACF uses this profile to deny access to a data set. If you specify NOTIFY without *userid*, RACF takes your user ID as the default; you are notified whenever the profile denies access to a data set.

A user who is to receive NOTIFY messages should log on frequently, both to take action in response to the unauthorized access attempts the messages describe and to clear the messages from the SYS1.BROADCAST data set. (When the profile also includes WARNING, RACF might have granted access to the data set to the user identified in the message.)

Note: The user ID specified on the NOTIFY operand is not notified when the profile disallows creation or deletion of a data set. NOTIFY is used only for resource access checking, not for resource creation or deletion.

OWNER(userid or group-name)

specifies a RACF-defined user or group to be assigned as the owner of the data set profile. When you define a group data set, the user you designate as owner must have at least USE authority in the group specified by the high-level qualifier of the data set name (or the qualifier determined by the naming conventions routine or by a command installation exit routine).

If you omit this operand, you are defined as the owner of the data set profile. However, if the high-level qualifier is a user ID that is different from your user ID, the OWNER of the profile is the user ID specified in the high-level qualifier. In addition, if you are using naming convention processing, either through the naming convention table or an exit, the owner of the profile is determined by the naming convention processing. If you have the SPECIAL attribute and define a profile for a group data set while SETROPTS ADDCREATOR is in effect, your user ID is added to the access list for the data set with ALTER access authority, whether or not you specify the OWNER operand. If you have the SPECIAL attribute and define a profile for a user data set, your user ID is not added to the access list for the data set.

If you specify OWNER(*userid*), the user you specify as the owner does not automatically have access to the data set. Use the PERMIT command to add the owner to the access list as desired. If you specify OWNER(*group-name*), RACF treats any users who have the group-SPECIAL attribute in the group as owners of the data set profile.

RETPD(*nnnnn*)

specifies the RACF security retention period for a tape data set. The security retention period is the number of days that must elapse before a tape data set profile expires. (Note that, even though the data set profile expires, RACF-protection for data sets protected by the profile is still in effect. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.)

The number you specify, *nnnnn* must be one to five digits in the range of 0 through 65533. To indicate a data set that never expires, specify *nnnnn* as 99999. When 99999 is used, the SETROPTS command stores it internally as 65534.

The RACF security retention period is the same as the data set retention period specified by the EXPDT/RETPD parameters on the JCL DD statement only when the data set profile is discrete and you do not modify the RACF security retention period.

When the TAPEVOL class is active, RACF checks the RACF security retention period before it allows a data set to be overwritten. RACF adds the number of days in the retention period to the creation date for the data set. If the result is less than the current date, RACF continues to protect the data set.

When the TAPEVOL class is not active, RACF ignores the RETPD operand.

If you omit RETPD and your installation has established a default security retention period (through the RETPD operand on the SETROPTS command), RACF uses the default. If you omit RETPD and your installation has not established a default, RACF uses 0 as a default.

Specifying this operand for a DASD data set does not cause an error, but it has no meaning because RACF ignores the operand during authorization checking.

SECLABEL(*security-label*)

specifies the user's default security label, where *security-label* is an installation-defined security label name that represents an association between a particular security level and a set of zero or more categories.

A security label corresponds to a particular security level (such as CONFIDENTIAL) with a set of zero or more security categories (such as PAYROLL or PERSONNEL).

RACF stores the name of the security label you specify in the data set profile if you are authorized to use that label.

If you are not authorized to use the security label or if the name you had specified is not defined as a SECLABEL profile in the SECLABEL class, the data set profile is not created.

SECLEVEL(*security-level*)

specifies the name of an installation-defined security level. This name corresponds to the number that is the minimum security level a user must have to access the data set. *security-level* must be a member of the SECLEVEL profile in the SECDATA class.

When you specify SECLEVEL and the SECDATA class is active, RACF adds security level access checking to its other authorization checking. If global access checking does not grant access, RACF compares the security level allowed in the user profile with the security level required in the data set profile. If the security level in the user profile is less than the security level in the data set profile, RACF denies the access. If the security level in the user profile is equal to or greater than the security level in the data set profile, RACF continues with other authorization checking.

Note: RACF does not perform security level checking for a started task or user that has the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF started procedures table or STARTED class, or to other users by customer-supplied RACF exits.

If the SECDATA class is not active, RACF still stores the *security-level* you specified in the data set profile, but cannot perform security level checking until you have activated the SECDATA class. If the name you specify is not defined as a SECLEVEL profile and the SECDATA class is active, you are prompted to provide a valid name for *security-level*.

TME

specifies that information for the Tivoli Security Management Application is to be added.

Note: The TME segment fields are intended to be updated only by the Tivoli Security Management application, which manages updates, permissions, and cross references. A security administrator should only directly update TME fields on an exception basis.

ROLES(*role-access-specification ...*)

specifies a list of roles and associated access levels related to this profile.

One or more *role-access-specification* values can be specified, each separated by blanks. Each value should contain no imbedded blanks and should have the following format:

role-name:access-authority[:conditional-class:conditional-profile]

where *role-name* is a discrete general resource profile defined in the ROLE class. The *access-authority* is the authority (NONE, EXECUTE, READ, UPDATE, CONTROL, or ALTER) with which groups in the role definition should be permitted to the resource.

The *conditional-class* is a class name (APPCPORT, CONSOLE, JESINPUT, PROGRAM, TERMINAL, or SYSID) for conditional access permission, and is followed by the *conditional-profile* value, a resource profile defined in the conditional class.

UACC(*access-authority*)

specifies the universal access authority to be associated with the data sets. The universal access authorities are ALTER, CONTROL, UPDATE, READ, EXECUTE, and NONE. If you omit UACC or specify UACC with no access authority, RACF uses the default value in your current connect group. If you specify CONTROL for a tape data set or a non-VSAM DASD data set, RACF treats the access authority as UPDATE. If you specify EXECUTE for a tape data set, or a DASD data set not used as a program library, RACF treats the access authority as NONE.

If a user accessing a data set has the RESTRICTED attribute, RACF treats the universal access authority (UACC) as NONE for that access attempt.

UNIT(*type*)

specifies the unit type on which a tape data set or a non-VSAM DASD data set resides. You can specify an installation-defined unit name, a generic device type, or a specific device address. If you specify UNIT and VOLUME for a DASD data set, RACF assumes that the data set is a non-VSAM data set; therefore, do not use UNIT and VOLUME for a VSAM data set.

If the data set is not cataloged, UNIT and VOLUME are required. You must specify UNIT and VOLUME for data sets cataloged with an esoteric name (such as an installation-defined unit name).

If you specify a generic or model profile name, RACF ignores this operand.

VOLUME(*volume-serial ...*)

specifies the volumes on which a tape data set or a non-VSAM DASD data set resides. If you specify UNIT and VOLUME for a DASD data set, RACF assumes that the data set is a non-VSAM data set; therefore, do not use UNIT and VOLUME for a VSAM data set.

If the data set is not cataloged, UNIT and VOLUME are required. You must specify UNIT and VOLUME for data sets cataloged with an esoteric name (such as an installation-defined unit name).

If you specify a tape data set profile name, you can specify only one volume.

If you specify a generic or model profile name, RACF ignores this operand.

WARNING

specifies that even if access authority is insufficient, RACF is to issue a warning message and allow access to the resource. RACF also records the access attempt in the SMF record if logging is specified in the profile.

Examples

Table 6. Examples of ADDSD

Example 1

<i>Operation</i>	User ADM1 wants to create a generic profile to protect all data sets having the high-level qualifier SALES. Only users with a security level of CONFIDENTIAL or higher are to be able to access the data sets.
<i>Known</i>	User ADM1 has the SPECIAL attribute and the installation has defined CONFIDENTIAL as a valid security level name. User ADM1 wants to issue the command as a RACF TSO command.
<i>Command</i>	ADDSD 'SALES.*' UACC(READ) AUDIT(ALL(READ)) SECLEVEL(CONFIDENTIAL)
<i>Defaults</i>	OWNER(ADM1) LEVEL(0)

ADDSD

Table 6. Examples of ADDSD (continued)

Example 2	<i>Operation</i>	User AEH0 wants to protect the data set AEH0.DEPT1.DATA with a discrete RACF profile.
	<i>Known</i>	User AEH0 is RACF-defined. AEH0.DEPT1.DATA is not cataloged. It resides on volume USER03 which is a 3330 volume. User AEH0 wants to issue the command as a RACF TSO command.
	<i>Command</i>	ADDSD 'AEH0.DEPT1.DATA' UNIT(3330) VOLUME(USER03)
	<i>Defaults</i>	OWNER(AEH0) UACC(UACC of user AEH0 in current connect group) AUDIT(FAILURES(READ)) LEVEL(0) SET
Example 3	<i>Operation</i>	User ADM1 wants to RACF-define the DASD data set SYS1.ICH02.DATA which was brought from another system where it was protected by a discrete RACF profile and was RACF-indicated. On the new system, only users with a security category of DEPT1 are to be allowed to access the data set.
	<i>Known</i>	User ADM1 has the SPECIAL attribute. SYS1.ICH02.DATA is cataloged. User ADM1 has create authority in group SYS1 and is connected to group SYS1 with the group-SPECIAL attribute. The installation has defined DEPT1 as a valid security category. User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	ADDSD 'SYS1.ICH02.DATA' OWNER(SYS1) UACC(NONE) AUDIT(ALL) NOSET CATEGORY(DEPT1)
	<i>Defaults</i>	LEVEL(0)
Example 4	<i>Operation</i>	User AEHO wants to create a model profile for group RSC and place an installation-defined description in the profile.
	<i>Known</i>	User AEHO has at least CREATE authority in group RSC. User AEHO wants to issue the command as a RACF TSO command.
	<i>Command</i>	ADDSD 'RSC.ACCESS.PROFILE' MODEL DATA('PROFILE THAT CONTAINS MODELING INFORMATION')
	<i>Defaults</i>	OWNER(AEHO), UACC(the UACC of user AEHO in current group) AUDIT(FAILURES(READ)) LEVEL(0)
Example 5	<i>Operation</i>	User AEH1 wants to protect the tape data set named AEH1.TAPE.RESULTS with a discrete RACF profile.
	<i>Known</i>	User AEH1 is a RACF-defined user. Data set AEH1.TAPE.RESULTS is cataloged, and tape data set protection is active. User AEH1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	ADDSD 'AEH1.TAPE.RESULTS' UACC(NONE) AUDIT(ALL(READ)) TAPE NOTIFY FILESEQ(1) RETPD(100)
	<i>Defaults</i>	LEVEL(0)
Example 6	<i>Operation</i>	User AEH1 wants to protect the tape data set named AEH1.TAPE.FUTURES with a discrete RACF profile, which is so much like the profile created for AEH1.TAPE.RESULTS (Example 5) that AEH1 can use the existing profile as a model for the new profile.
	<i>Known</i>	User AEH1 is a RACF-defined user. Data set AEH1.TAPE.FUTURES is cataloged, and tape data set protection is active. User AEH1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	ADDSD 'AEH1.TAPE.FUTURES' FROM('AEH1.TAPE.RESULTS') FILESEQ(2)
	<i>Defaults</i>	LEVEL(0)

Table 6. Examples of ADDSD (continued)

Example 7

Operation User ADM1 wants to create a generic profile to protect all data sets having the high-level qualifier PROJECTA. The data sets protected by the profile will be managed by DFP. Group TEST4 will be assigned as the actual owner of the data sets protected by the profile. The profile will have a universal access authority of READ.

User ADM1 wants to direct the command to run at the local node under the authority of user DAP02 and prohibit the command from being automatically directed to other nodes.

Known Users ADM1 and DAP02 have the SPECIAL attribute, the operating system has DFP 3.1.0 installed, and TEST4 is a RACF-defined group. Users ADM1 and DAP02 have an already established user ID association. User ADM1 wants to issue the command as a RACF TSO command.

Command ADDSD 'PROJECTA.*' UACC(READ) DFP(RESOWNER(TEST4)) ONLYAT(.DAP02)

Defaults OWNER(ADM1) LEVEL(0) AUDIT(FAILURES(READ))

Results The command is only processed on the local node and not automatically directed to any other nodes in the RRSF configuration.

Example 8

Operation User TSO7 wants to create a generic profile to protect all data sets having the high-level qualifier PROJECTB with a security label of CONF. User TSO7 is authorized to the security label. User TSO7 wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.

Known User TSO7 is a RACF-defined user.

Command @ADDSD 'PROJECTB.*' SECLABEL(CONF)

Defaults None

ADDUSER (Add User Profile)

Purpose

Use the ADDUSER command to define a new user to RACF and establish the user's relationship to an existing RACF-defined group.

The command adds a profile for the new user to the RACF database and creates a connect profile that connects the user to whichever default group you specify.

The user profile consists of a RACF segment and, optionally, other segments such as a TSO segment, a DFP segment, or an OMVS segment. You can use this command to define information in any segment of the user's profile.

Although user ID association information is in the user's profile, you must use the RACLINK command to define a user ID association.

Attention:

- When the ADDUSER command is issued from ISPF, the TSO command buffer (including password data) is written to the ISPLOG data set. As a result, you should not issue this command from ISPF or you must control the ISPLOG data set carefully.
- If the ADDUSER command is issued as a RACF operator command, the command and all data (including password data) is written to the system log. You should not issue the ADDUSER command as an operator command unless specifying PROTECTED. For all other cases you should execute it as a TSO command.

This command is not intended to be used for profiles in the DIGTCERT or DIGTNMAP classes.

Issuing Options

The following table identifies the eligible options for issuing the ADDUSER command:

Table 7. How the ADDUSER Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	Yes	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To change a user profile, see “ALTUSER (Alter User Profile)” on page 115.
- To delete a user profile, see “DELUSER (Delete User Profile)” on page 189.
- To list a user profile, see “LISTUSER (List User Profile)” on page 223.

- To administer user ID associations, see “RACLINK (Administer User ID Associations)” on page 296.
- To obtain a list of user profiles, see “SEARCH (Search RACF Database)” on page 408.

Authorization Required

When issuing the ADDUSER command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

To use the ADDUSER command, you must have one of the following:

- The SPECIAL attribute
- The CLAUTH attribute for the USER class while one of the following is true:
 - You are the owner of the default group specified in this command.
 - You have JOIN authority in the default group specified in this command.
 - The default group is within the scope of a group in which you have the group-SPECIAL attribute.

You must have the SPECIAL attribute to give the new user the OPERATIONS, SPECIAL, or AUDITOR attribute. You need not, however, have the SPECIAL attribute to specify the OWNER operand.

You cannot assign a user an attribute or authority higher than your own.

To assign a security category to a profile, you must have the SPECIAL attribute, or the category must be in your user profile. To assign a security level to a profile, you must have the SPECIAL attribute or, in your own profile, a security level that is equal to or greater than the security level you are assigning.

To assign a security label, you must have the SPECIAL attribute or have READ authority to the security label profile. However, the security administrator can limit the ability to assign security labels only to users with the SPECIAL attribute.

To define information within a segment other than the base segment, you must have one of the following:

- The SPECIAL attribute
- At least UPDATE authority to the desired field within the segment through field-level access control

For information on field-level access checking, see *z/OS Security Server RACF Security Administrator's Guide*.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

To specify the SHARED keyword, you must have the SPECIAL attribute or at least READ authority to the SHARED.IDS resource in the UNIXPRIV class.

ADDUSER

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the ADDUSER command is:

```
[subsystem-prefix]{ADDUSER | AU}
    (userid ...)
    [ ADDCATEGORY(category-name ...) ]
    [ ADSP | NOADSP ]
    [ AT([node].userid ...) | ONLYAT([node].userid ...) ]
    [ AUDITOR | NOAUDITOR ]
    [ AUTHORITY(group-authority) ]
    [ CICS(
        [ OPCLASS(operator-class ...) ]
        [ OPIDENT(operator-id) ]
        [ OPPRTY(operator-priority) ]
        [ TIMEOUT(timeout-value) ]
        [ XRFSSOFF( FORCE | NOFORCE ) ]
    ) ]
    [ CLAUTH(class-name ...) | NOCLAUTH ]
    [ DATA('installation-defined-data') ]
    [ DCE(
        [ AUTOLOGIN(YES|NO) ]
        [ DCENAME(user-principal-name) ]
        [ HOMECCELL(dce-cell-name) ]
        [ HOMEUUID(home-cell-universal-unique-identifier) ]
        [ UUID(universal-unique-identifier) ]
    ) ]
    [ DFLTGRP(group-name) ]
    [ DFP(
        [ DATAAPPL(application-name) ]
        [ DATACLAS(data-class-name) ]
        [ MGMTCLAS(management-class-name) ]
        [ STORCLAS(storage-class-name) ]
    ) ]
    [ EIM ( LDAPPROF ( ldapbind_profile ) ) ]
    [ GRPACC | NOGRPACC ]
    [ KERB (
        [ ENCRYPT (
            [ DES | NODES ]
            [ DES3 | NODES3 ]
            [ DESD | NODESD ]
        ) ]
        [ KERBNAME(kerberos-principal-name) ]
        [ MAXTKTLFE(max-ticket-life) ]
    ) ]
    [ LANGUAGE(
        [ PRIMARY(language) ]
        [ SECONDARY(language) ]
    ) ]
    [ LNOTES(
        [ SNAME(short-name) ]
    ) ]
    [ MODEL(dsname) ]
    [ NAME(user-name) ]
    [ NDS(
        [ UNAME(user-name) ]
    ) ]
```

```

[ NETVIEW(
  [ CONSNAME(console-name) ]
  [ CTL(GENERAL | GLOBAL | SPECIFIC) ]
  [ DOMAINS(domain-name ...) ]
  [ IC('command | command-list') ]
  [ MSGRECVR(NO | YES) ]
  [ NGMFADMN(NO | YES) ]
  [ NGMFVSPN(view-span) ]
  [ OPCLASS(class ... ) ]
) ]
[ OIDCARD | NOOIDCARD ]
[OMVS [ (
  [ ASSIZEMAX(address-space-size)]
  [ AUTOUID | UID ( user-identifier ) [ SHARED ] ]
  [ CPUTIMEMAX(cpu-time) ]
  [ FILEPROCMAX(files-per-process) ]
  [ HOME(initial-directory-name) ]
  [ MMAPAREAMAX(memory-map-size) ]
  [ PROCUSERMAX(processes-per-UID) ]
  [ PROGRAM(program-name) ]
  [ THREADSMAX(threads-per-process) ]
) ] ]
[ OPERATIONS | NOOPERATIONS ]
[ OPERPARM( [ ALTGRP(alternate-console-group) ]
  [ AUTH(operator-authority) ]
  [ AUTO( YES | NO ) ]
  [ CMDSYS(system-name) ]
  [ DOM( NORMAL | ALL | NONE ) ]
  [ KEY(searching-key) ]
  [ LEVEL(message-level) ]
  [ LOGCMDRESP( SYSTEM | NO ) ]
  [ MFORM(message-format) ]
  [ MIGID( YES | NO ) ]
  [ MONITOR(event) ]
  [ MSCOPE( system-names | * | *ALL ) ]
  [ ROUTCODE( ALL | NONE | routing-codes ) ]
  [ STORAGE(amount) ]
  [ UD( YES | NO ) ]
) ]
[OVM(
  [ FSROOT(file-system-root) ]
  [ HOME(initial-directory-name) ]
  [ PROGRAM(program-name) ]
  [ UID(user-identifier) ]
) ]
[ OWNER(userid or group-name) ]
[ PASSWORD(password) | NOPASSWORD ]
[PROXY[(
  [ LDAPHOST(ldap_url) ]
  [ BINDDN(bind_distinguished_name) ]
  [ BINDPW(bind_password) ] ) ] ]
[ RESTRICTED | NORESTRICTED ]
[ SECLABEL(seclabel-name) ]
[ SECLEVEL(seclevel-name) ]
[ SPECIAL | NOSPECIAL ]

```

ADDUSER

```
[TSO(
  [ ACCTNUM(account-number) ]
  [ COMMAND(command-issued-at-logon) ]
  [ DEST(destination-id) ]
  [ HOLDCLASS(hold-class) ]
  [ JOBCLASS(job-class) ]
  [ MAXSIZE(maximum-region-size) ]
  [ MSGCLASS(message-class) ]
  [ PROC(logon-procedure-name) ]
  [ SECLABEL(security-label) ]
  [ SIZE(default-region-size) ]
  [ SYSOUTCLASS(sysout-class) ]
  [ UNIT(unit-name) ]
  [ USERDATA(user-data) ]
) ]
[ UACC(access-authority) ]
[ WHEN( [DAYS(day-info)] [TIME(time-info)] ) ]
[ WORKATTR(
  [ WAACNT(account-number) ]
  [ WAADDR1(address-line-1) ]
  [ WAADDR2(address-line-2) ]
  [ WAADDR3(address-line-3) ]
  [ WAADDR4(address-line-4) ]
  [ WABLDG(building) ]
  [ WADEPT(department) ]
  [ WANAME(name) ]
  [ WAROOM(room) ]
) ]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

userid

Specifies the user to be defined to RACF. If you are defining more than one user, the list of user IDs must be enclosed in parentheses.

This operand is required and must be the first operand following ADDUSER.

Each user ID must be unique and must not currently exist on the RACF database as a user ID or a group name.

ADDCATEGORY(*category-name ...*)

Specifies one or more names of installation-defined security categories. The names you specify must be defined as members of the CATEGORY profile in a SECDATA class. (For information on defining security categories, see *z/OS Security Server RACF Security Administrator's Guide*.)

When the SECDATA class is active and you specify ADDCATEGORY, RACF performs security category checking in addition to its other authorization checking. If a user requests access to a resource, RACF compares the list of security categories in the user's profile with the list of security categories in the resource profile. If RACF finds any security category in the resource profile that is not in the user's profile, RACF denies access to the resource. If the user's profile contains all the required security categories, RACF continues with other authorization checking.

Note: RACF does not perform security category checking for a started task or user with the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF started procedures table or STARTED class, or to other users by customer-supplied RACF exits.

ADSP | NOADSP**ADSP**

Specifies that all permanent tape and DASD data sets created by the new user are automatically be RACF-protected by discrete profiles. ADSP specified on the ADDUSER command overrides NOADSP specified on the CONNECT command.

If SETROPTS NOADSP is in effect, RACF ignores the ADSP attribute at logon or job initiation.

NOADSP

Specifies that the new user is not to have the ADSP attribute. NOADSP is the default value if you omit both ADSP and NOADSP.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([*node*].*userid ...*)

Specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT([*node*].*userid ...*)

Specifies that the command is to be directed only to the node specified by *node* where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed only to the local node.

AUDITOR | NOAUDITOR**AUDITOR**

Specifies that the new user has full responsibility for auditing the use of system resources, and is able to control the logging of detected accesses to any RACF-protected resources during RACF authorization checking and accesses to the RACF database.

ADDUSER

You must have the SPECIAL attribute to enter the AUDITOR operand.

NOAUDITOR

Specifies that the new user does not have the AUDITOR attribute. NOAUDITOR is the default value if you omit both AUDITOR and NOAUDITOR.

AUTHORITY(*group-authority*)

Specifies the level of group authority for the new user in the default group. The valid group authority values are USE, CREATE, CONNECT, and JOIN and are described in “Group authorities” on page 13. If you omit this operand or specify AUTHORITY without *group-authority*, the default value is USE.

This operand is group-related. If a user is connected to other groups (with the CONNECT command), the user can have a different group authority in each group.

CICS

Defines CICS operator information for a new CICS terminal user. This operand requires CICS/ESA* 3.2.1 or later.

You can control access to an entire CICS segment or to individual fields within the CICS segment by using field-level access checking. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

OPCLASS(*operator-class ...*)

Specifies numbers in the range of 1 through 24, defined as two digits, representing classes assigned to this operator to which BMS (basic mapping support) messages are to be routed.

OPIDENT(*operator-id*)

Specifies a 1-3 character identification of the operator for use by BMS.

Operator identifiers can consist of any characters, and can be entered with or without single quotes. The following rules apply:

- If parentheses, commas, blanks, or semicolons are to be entered as part of the operator identifier, the character string must be enclosed in single quotes. For example, if the operator identifier is (1), you must enter OPIDENT(' (1) ').
- If a single quote is intended to be part of the character string, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

If OPIDENT is not specified, the field defaults to blanks in the RACF user profile, and blanks appear in the field in the LISTUSER command output.

OPPRTY(*operator-priority*)

Specifies the number in the range of 0 through 255 that represents the priority of the operator.

If OPPRTY is not specified, the field defaults to zeros in the RACF user profile, and zeros appear in the field in the LISTUSER command output.

TIMEOUT(*timeout-value*)

Specifies the time, in hours and minutes, that the operator is allowed to be idle before being signed off. The value for TIMEOUT can be entered in the form *m*, *mm*, *hmm*, *hhmm*, where the value for *m* or *mm* can range from 00 to 59, or 00 to 60 if *h* or *hh* is not specified or specified as 0 or 00. The value for *h* or *hh* can range from 00 to 99.

Note: If your installation has multiple levels of CICS sharing the RACF database, be aware that versions of CICS prior to CICS 4.1 allow TIMEOUT values only in the form *mm* where *mm* is 0 to 60. A specification of 0200 is interpreted as two hours on a CICS 4.1 system, but it causes an earlier CICS system to see a timeout value of 0 and assume no timeout is to be used. To avoid this problem, users with mixed CICS systems might wish to specify a timeout value in the form *hhmm* and ensure the *mm* value is not zero. For example, you could specify a TIMEOUT value such as 0159, which is interpreted as 1 hour and 59 minutes on a CICS 4.1 system, and is interpreted as 59 minutes on an earlier CICS system.

TIMEOUT defaults to 0 if omitted, meaning no timeout.

XRFSOFF(FORCE | NOFORCE)

FORCE means that the user is signed off by CICS when an XRF takeover occurs.

CLAUTH | NOCLAUTH

CLAUTH(*class-name ...*)

Specifies the classes in which the new user is allowed to define profiles to RACF for protection. Classes you can specify are USER, and any resource classes defined in the class descriptor table.

To enter the CLAUTH operand, you must have the SPECIAL attribute or have the CLAUTH attribute for the classes specified. If you do not have sufficient authority for a specified class, RACF ignores the CLAUTH specification for that class and continues processing with the next class name specified.

Note: The CLAUTH attribute has no meaning for the FILE and DIRECTORY classes.

NOCLAUTH

Specifies that the new user is not to have the CLAUTH attribute. NOCLAUTH is the default if you omit both CLAUTH and NOCLAUTH.

DATA('installation-defined-data')

Specifies up to 255 characters of installation-defined data to be stored in the user's profile and must be enclosed in quotes. It can also contain double-byte character set (DBCS) data. Note that only 254 characters are chained off the ACEE.

Use the LISTUSER command to list this information.

DCE

Specifies that, when you define an z/OS DCE user to RACF, you can enter any of the following suboperands to specify information for that user. Each suboperand defines information that RACF stores in a field within the DCE segment of the user's profile.

You can control access to an entire DCE segment or to individual fields within the DCE segment by using field level access checking.

To define information within the DCE segment, you must have one of the following:

- The SPECIAL attribute
- At least UPDATE authority to the desired field within the segment through field-level access control

ADDUSER

For information on field-level access checking, see *z/OS Security Server RACF Security Administrator's Guide*.

Attention

The ability to associate a RACF and DCE identity depends on replicated information between DCE and RACF. You should **not** change the user's UUID, principal name, or cell name in either RACF or the DCE registry without a corresponding update in the other registry.

AUTOLOGIN(NO | YES)

Specifies whether z/OS UNIX DCE is to log this user into z/OS UNIX DCE automatically. If AUTOLOGIN(NO) is specified, z/OS UNIX DCE does *not* attempt to login this user to z/OS UNIX DCE automatically. If AUTOLOGIN is not specified, AUTOLOGIN(NO) is the default.

DCENAME(*user-principal-name*)

Specifies the DCE principal name defined for this RACF user in the DCE registry.

The DCENAME you define to RACF can contain 1-1023 characters and can consist of any character. You can enter the name with or without single quotes, depending on the following:

- If parentheses, commas, blanks, or semicolons are entered as part of the name, the character string must be enclosed in single quotes.
- If a single quote is intended to be part of the name, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered. RACF does not ensure that a valid DCENAME has been specified.

The DCENAME assigned to a user must be the same as the DCE principal name defined to the DCE registry.

If DCENAME is not specified, the user cannot login as an z/OS UNIX DCE user automatically, even when AUTOLOGIN(YES) is specified.

Note: RACF does not enforce the uniqueness of each DCENAME. The DCENAME specified must match the user's DCE principal name that is defined to the DCE registry. If the DCENAME entered does not correspond to the DCE principal name entered in the DCE registry for this user, z/OS UNIX DCE cannot correctly associate the identity of the DCE principal with the correct RACF user ID.

HOMECELL(*dce-cell-name*)

Specifies the DCE cell name defined for this RACF user.

The HOMECELL you define to RACF can contain 1-1023 characters and can consist of any character. You can enter the name with or without single quotes, depending on the following:

- If parentheses, commas, blanks, or semicolons are entered as part of the cell name, the character string must be enclosed in single quotes.
- If a single quote is intended to be part of the cell name, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered. The fully qualified path name should be specified. RACF does not ensure that a valid DCE cell name has been specified.

The HOMECCELL assigned to a user **must** be the same as the DCE cell name that this user has been defined to.

If the HOMECCELL is not specified, z/OS UNIX DCE single signon to DCE support assumes that the HOMECCELL for this user is the same cell that this MVS system is defined to.

RACF checks that the prefix of the HOMECCELL name entered has a prefix of either */.../* or */./*.

The notation */.../* indicates that the HOMECCELL name is a global domain name service (DNS) cell name or X.500 global name.

The notation */./* indicates that the HOMECCELL name is a cell relative CDS (cell directory service) name. When determining the naming conventions used within your DCE cell, you should contact your DCE cell administrator.

HOMEUUID(*home-cell-universal-unique-identifier*)

Specifies the DCE universal unique identifier (UUID) for the cell that this user is defined to. The UUID is a 36-character string that consists of numeric and hexadecimal characters. This string must have the delimiter of "-" in character positions 9, 14, 19, and 24. The general format for the UUID string is xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx, in which x represents a valid numeric or hexadecimal character.

Be careful when assigning UUIDs. The UUID *cannot* be randomly assigned. The HOMEUUID is the DCE UUID of the cell that this RACF user is defined to. If HOMEUUID is not specified, the LISTUSER command displays NONE for the HOMEUUID field.

Note: The HOMEUUID specified must match the UUID of the DCE cell to which this principal (specified by the DCENAME operand) is defined.

UUID(*universal-unique-identifier*)

Specifies the DCE universal unique identifier (UUID) of the DCE principal defined in DCENAME. The UUID is a 36-character string that consists of numeric and hexadecimal characters. This string must have the delimiter of "-" in character positions 9, 14, 19, and 24. The general format for the UUID string is xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx, in which x represents a valid numeric or hexadecimal character.

Be careful when assigning UUIDs. The UUID *cannot* be randomly assigned. Note that RACF does not enforce the uniqueness of each UUID entered. The DCE UUID assigned to a user **must** be the same as the DCE UUID assigned when defining this RACF user to the DCE registry as a DCE principal that is being cross-linked with this RACF user ID. This DCE principal is specified using the DCENAME operand.

DFLTGRP(*group-name*)

Specifies the name of a RACF-defined group to be used as the default group for the user. If you do not specify a group, RACF uses your current connect group as the default.

ADDUSER

Note: You do not have to issue the CONNECT command to connect new users to their default groups.

DFP

Specifies that when you define a user to RACF, you can enter any of the following suboperands to specify default values for DFP data application identifier, data class, management class, and storage class. DFP uses this information to determine data management and DASD storage characteristics when a user creates a new data set.

You can control access to an entire DFP segment or to individual fields within the DFP segment by using field-level access checking. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

DATAAPPL(*application-name*)

specifies an 8-character DFP data application identifier.

DATACLAS(*data-class-name*)

Specifies the default data class. The maximum length of *data-class-name* is 8 characters.

A data class can specify some or all of the physical data set attributes associated with a new data set. During new data set allocation, data management uses the value you specify as a default unless it is preempted by a higher priority default, or overridden in some other way, for example by JCL.

Note: The value you specify must be a valid data class name defined for use on your system. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

For information on defining DFP data classes, see *z/OS DFSMSdfp Storage Administration Reference*.

MGMTCLAS(*management-class-name*)

Specifies the default management class. The maximum length of *management-class-name* is 8 characters.

A management class contains a collection of management policies that apply to data sets. Data management uses the value you specify as a default unless it is preempted by a higher priority default, or overridden in some other way, for example by JCL.

Note: The value you specify must be protected by a profile in the MGMTCLAS general resource class, and the user must be granted at least READ access to the profile. Otherwise, RACF does not allow the user access to the specified MGMTCLAS. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

For information on defining DFP management classes, see *z/OS DFSMSdfp Storage Administration Reference*.

STORCLAS(*storage-class-name*)

Specifies the default storage class. The maximum length of *storage-class-name* is 8 characters.

A storage class specifies the service level (performance and availability) for data sets managed by the storage management subsystem (SMS). During new data set allocation, data management uses the value you specify as a

default unless it is preempted by a higher priority default, or overridden in some other way (for example, by JCL).

Note: The value you specify must be protected by a profile in the STORCLAS general resource class, and the user must be granted at least READ access to the profile. Otherwise, RACF does not allow the user access to the specified STORCLAS. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

For information on defining DFP storage classes, see *z/OS DFSMSdfp Storage Administration Reference*.

EIM

Specifies the bind information required to establish a connection with the EIM domain.

LDAPPROF(*ldapbind_profile*)

Specifies the name of a profile in the LDAPBIND class. The profile in the LDAPBIND class contains the name of an EIM domain and the bind information required to establish a connection with the EIM domain. The EIM services will attempt to retrieve this information when it is not explicitly supplied via invocation parameters. Applications or other services that use the EIM services may instruct their invokers to define a profile in the LDAPBIND class or the IRR.PROXY.DEFAULTS profile in the FACILITY class.

GRPACC | NOGRPACC

GRPACC

Specifies that any group data sets protected by DATASET profiles defined by the new user are automatically accessible to other users in the group. The group whose name is used as the high-level qualifier of the data set name (or the qualifier supplied by a command installation exit) has UPDATE access authority in the new profile. GRPACC specified on the ADDUSER command overrides NOGRPACC specified on the CONNECT command.

NOGRPACC

Specifies that the new user does not have the GRPACC attribute. NOGRPACC is the default value if you omit both GRPACC and NOGRPACC.

KERB

Specifies Security Server Network Authentication Service information for a user you are defining to RACF. Each subkeyword defines information that RACF stores in a field within the KERB segment of the user's profile.

Note: The RACF user password must be changed to be non-expired in order to complete the definition of the Network Authentication Service principal. The user cannot use any Network Authentication Service function until the definition is complete.

ENCRYPT [[DES] NODES] [DES3 | NODES3] [DESD | NODESD]]

The ENCRYPT values are used to specify which keys are allowed for use based on the encryption algorithm used to generate them. The default values for ENCRYPT are DES, DES3, and DESD. You can use the following values to specify which keys are allowed for use by a principal.

DES	DES encrypted keys are allowed for use.
NODES	No DES encrypted keys are allowed for use.
DES3	DES3 encrypted keys are allowed for use.

ADDUSER

NODES3	No DES3 encrypted keys are allowed for use.
DESD	DESD encrypted keys are allowed for use.
NODESD	No DESD encrypted keys are allowed for use.

The values in effect are dependent on the current SETROPTS KERBLVL setting.

When SETROPTS KERBLVL(0) is in effect, the ENCRYPT settings will be ignored. Regardless of the settings, DES keys will be generated and processed.

When SETROPTS KERBLVL(1) is in effect, or when SETROPTS KERBLVL gets changed from 0 to 1, the ENCRYPT settings will go into effect. Therefore, on password change, all three keys are generated and stored in the user's profile. The ENCRYPT setting will be used to determine which keys can be processed.

If you do not want to accept the defaults, you must specify the values you desire. For example, if you want to use only DES3 encryption, you must specify ENCRYPT(NODES DES3 NODESD).

If you specify ENCRYPT(NODES, NODES3, NODESD) at KERBLVL(1), no keys can be used, but all three will be generated and stored. At KERBLVL(0), the DES key will still be generated and it cannot be disallowed.

KERBNAME(*kerberos-principal-name*)

Specifies the z/OS user ID's local *kerberos-principal-name*.

The value specified for the local *kerberos-principal-name* must be unique. Consequently, a list of users cannot be specified on an ADDUSER command with the KERBNAME keyword.

The *kerberos-principal-name* you define to RACF can consist of any character except the @ (X'7C') character. It is highly recommended that you avoid using **any** of the EBCDIC variant characters be avoided to prevent problems between different code pages. You can enter the name with or without single quotes, depending on the following:

- If parentheses, commas, blanks, or semicolons are entered as part of the name, the name must be enclosed in single quotes.
- If a single quote is intended to be part of the name and the entire character string is enclosed in single quotes, you must use two single quotes together to represent each single quote within the string.
- If the first character of the name is a single quote, you must enter the string within single quotes, with two single quotes entered for that single quote.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. However, RACF does not ensure that a valid *kerberos-principal-name* has been specified.

A local *kerberos-principal-name* **must not** be qualified with a realm name when specified with the KERBNAME keyword. However, RACF verifies that the local principal name, when fully qualified with the name of the local realm:

`/.../local_realm_name/principal_name`

does not exceed 240 characters. For example,

- If the local realm name is

X

fully qualified local principal names are prefixed with

/.../X/

and are limited to a maximum of 233 characters.

- If the local realm name is

KERB390.ENDICOTT.IBM.COM

fully qualified local principal names will be prefixed with

/.../KERB390.ENDICOTT.IBM.COM/

and be limited to a maximum of 210 characters.

This length verification requires that the REALM profile for the local realm KERBDFTL be defined and contain the name of the local realm, prior to the specification of local Network Authentication Service principals. Otherwise, Network Authentication Service users might not be properly defined.

Note: Because of the relationship between local realm names and local *kerberos-principal-names*, in which the length of a fully qualified name cannot exceed 240 characters, caution and planning must go into renaming the local realm since the combined length is only checked by RACF when a local *kerberos-principal-name* is added or altered. Renaming the realm should be avoided as a result.

MAXTKLFE(*max-ticket-life*)

Specifies the *max-ticket-life* in seconds, and is represented by a numeric value between 1 and 2 147 483 647. Note that 0 is not a valid value.

If MAXTKLFE is specified on the definition of a local Network Authentication Service principal, the Security Server Network Authentication Service takes the most restrictive of the value defined for the local principal and the value specified on the definition of the local realm (the KERBDFTL profile in the REALM class). Consequently, if the realm *max-ticket-life* is 24 hours, a principal cannot get a ticket with a longer lifetime even if the *max-ticket-life* is set to 48 hours in the user profile. If this field is not specified for a local principal, or if NOMAXTKLFE has been specified, the maximum lifetime for tickets created for this principal is determined from the definition of the local Network Authentication Service realm.

LANGUAGE

Specifies the user's preferred national languages. Specify this operand if the user is to have languages other than the system-wide defaults (established by the LANGUAGE operand on the SETROPTS command).

- If this profile is for a TSO/E user who is to establish an extended MCS console session, the languages you specify should be one of the languages specified on the LANGUAGE LANGCODE statements in the MMSLSTxx PARMLIB member. See your MVS system programmer for this information. For more information on TSO/E national language support, see *z/OS TSO/E Customization*.

- If this profile is for a CICS user, see your CICS administrator for the languages supported by CICS on your system.

For more information, see *CICS RACF Security Guide*.

ADDUSER

PRIMARY(*language*)

Specifies the user's primary language.

SECONDARY(*language*)

Specifies the user's secondary language.

Notes:

1. For the primary and secondary languages, specify either the installation-defined name of a currently active language (a maximum of 24 characters) or one of the language codes (three characters in length) for a language installed on your system.
2. The language name can be a quoted or unquoted string.
3. The same language can be specified with both PRIMARY and SECONDARY parameters.
4. If the MVS message service is not active, the PRIMARY and SECONDARY values must be a 3-character language code.

LNOTES

Specifies the Lotus Notes for z/OS information for the user profile being added.

SNAME(*short-name*)

Specifies the Lotus Notes for z/OS *short-name* of the user being defined. This value should match the name stored in the Lotus® Notes™ for z/OS address book for this user, but this is not verified by the command.

The *short-name* you define to RACF can contain 1-64 characters. The *short-name* can contain the following characters: upper and lower case alphabetics (A-Z, a-z), 0 through 9, & (X'50'), - (X'60'), . (X'4B'), _ (X'6D'), and (X'40').

If the *short-name* you specify contains any blanks, it must be enclosed in single quotes. The *short-name* is stripped of leading and trailing blanks.

The value specified for the *short-name* must be unique. Consequently, a list of users cannot be specified on an ADDUSER command with the SNAME keyword.

MODEL(*dsname*)

Specifies the name of a discrete data set profile that is used as a model when new data set profiles are created that have *userid* as the high-level qualifier. For this operand to be effective, the MODEL(USER) option (specified on the SETROPTS command) must be active.

RACF always prefixes the data set name with *userid* when it accesses the model. For information about automatic profile modeling, refer to *z/OS Security Server RACF Security Administrator's Guide*.

NAME(*user-name*)

Specifies the user name to be associated with the new user ID. You can use a maximum of 20 alphanumeric or non-alphanumeric characters. If the name you specify contains any blanks, it must be enclosed in single quotes.

If you omit the NAME operand, RACF uses a default of 20 # (X'7B') characters ('###...'). Note, however, that the corresponding entry in a LISTUSER output is the word "unknown".

NDS

Specifies the Novell Directory Services for OS/390 information for the user profile being added.

UNAME(*user-name*)

Specifies the Novell Directory Services for OS/390 *user-name* of the user being defined. The *user-name* value should match the name stored in the Novell Directory Services for OS/390 directory for this user, but this is not verified by the command.

The *user-name* you define to RACF can contain 1-246 characters. However, the *user-name* can not contain the following characters: * (X'5C'), + (X'4E'), | (X'4F'), = (X'7E'), , (X'6B'), " (X'7F'), ` (X'79'), / (X'61'), : (X'7A'), ; (X'5E'), ¢ (X'4A'), and brackets ([and], X'AD' and X'BD' respectively).

If the *user-name* you specify contains any parentheses or blanks, it must be enclosed in single quotes. The *user-name* is stripped of leading and trailing blanks. If a single quotation mark is intended to be part of the *user-name*, you must use two single quotation marks together for each single quotation mark within the string, and the entire string must then be enclosed within single quotation marks.

The value specified for the *user-name* must be unique. Consequently, a list of users cannot be specified on an ADDUSER command with the UNAME keyword.

NETVIEW**CONSNAME(*console-name*)**

Specifies the default Master Console Station (MCS) console name used for this operator. This default console name is used when the operator does not specify a console name on the NetView GETCONID command.

Console-name is a 1-8 character identifier whose validity is checked by MVS processing when the operator tries to use it. See *z/OS MVS Planning: Operations* for information on valid values for a particular release.

CTL(GENERAL | GLOBAL | SPECIFIC)

Specifies whether a security check is performed for this NetView operator when they try to use a span or try to do a cross-domain logon.

GENERAL

Specifies that checking is done as described for SPECIFIC, and, in addition, that the operator is allowed to access devices that are not part of any span.

GLOBAL

Specifies that no checking is done.

SPECIFIC

Specifies that the operator is allowed to control only devices that are in spans the operator started, and that a security check is to be performed through RACROUTE REQUEST=AUTH whenever this operator attempts to use a span. Also, any cross-domain logon must be to a domain listed in the operator's NETVIEW segment with the DOMAINS keyword.

SPECIFIC is the default.

DOMAINS(*domain-name ...*)

Specifies the identifiers of NetView programs in another NetView domain where this operator can start a cross-domain session. The NetView program identifiers are coded on the NCCFID definition statement for the other domains, and represent the name given to that NetView program on the APPL statement.

ADDUSER

Domain-name is a 1-5 character identifier. The characters can be alphabetic, numeric, or national.

IC('command | command-list;')

Specifies the command or command list (up to 255 characters) to be processed by NetView for this operator when this operator logs on to NetView.

If the command or command list you specify contains any commas, blanks, or other special characters that TSO/E requires to be quoted, it must be enclosed in single quotes.

MSGRECVR(NOYES)

Specifies whether this operator is to receive unsolicited messages that are not routed to a specific NetView operator.

NO

Specifies that the operator is not to receive the messages.

NO is the default.

YES

Specifies that the operator is to receive the messages.

NGMFADMN(NOYES)

Specifies whether a NetView operator has administrator authority to the NetView Graphic Monitor Facility (NGMF).

NO

Specifies that the operator does not have authority.

NO is the default.

YES

specifies that the operator has the authority.

NGMFVSPN (*view-span*)

Reserved for future use by the NetView Graphic Monitor Facility

OPCLASS(*class ...*)

NetView scope classes for which the operator has authority. The OPCLASS values are only used if NetView is doing the checking itself, rather than using SAF and the NETCMDS class that RACF provides. If the OPCLASS operand is not specified, the operator is considered to have authority in scope classes.

class is a number from 1 to 2040 that specifies a NetView scope class.

OIDCARD | NOOIDCARD

OIDCARD

Specifies that the new user must supply an operator identification card when logging onto the system. If you specify the OIDCARD operand, the system prompts you to enter the new user's operator identification card as part of the processing of the ADDUSER command. If you specify the OIDCARD operand in a job executing in the background or when you cannot be prompted in the foreground, the ADDUSER command fails.

NOOIDCARD

Specifies that the new user is not required to supply an operator identification card. NOOIDCARD is the default value if you omit both OIDCARD and NOOIDCARD.

OMVS

Specifies z/OS UNIX System Services information for the user being defined to RACF. Information is stored in the OMVS segment of the user's profile.

You can control access to an entire OMVS segment or to individual fields in the OMVS segment by using field-level access checking.

ASSIZEMAX(*address-space-size*)

Specifies the RLIMIT_AS hard limit (maximum) resource value that processes receive when they are dubbed a process. The *address-space-size* you define to RACF is a numeric value between 10 485 760 and 2 147 483 647. ASSIZEMAX indicates the address space region size in bytes. The soft limit (current) resource value is obtained from MVS. If the soft limit value from MVS is greater than the *address-space-size*, the soft limit is used.

The value specified for ASSIZEMAX is also used when processes are initiated by a daemon process using an exec after `setuid()`. In this case, both the RLIMIT_AS hard and soft limits are set to the *address-space-size* value.

The value specified for ASSIZEMAX overrides any value provided by the MAXASSIZE parameter of BPXPRMxx. For more information, see *z/OS UNIX System Services Planning*.

AUTOUID I UID

Specifies whether RACF is to automatically assign an unused UID value to the user or if a specific UID value is to be assigned.

AUTOUID

Specifies that RACF is to automatically assign an unused UID value to the user. The UID value is derived from information obtained from the BPX.NEXT.USER profile in the FACILITY class. For more information on setting up BPX.NEXT.USER, see *z/OS Security Server RACF Security Administrator's Guide*.

If you are using RRSF automatic command direction for the USER class, the command sent to other nodes will contain an explicit assignment of the UID value which was derived by RACF on the local node.

Rules:

- AUTOUID cannot be specified if more than one user ID is entered.
- The AUTOUID keyword is mutually exclusive with the SHARED keyword.
- If both UID and AUTOUID are specified, AUTOUID is ignored.
- Field-level access checking for the UID field applies when using AUTOUID.

UID(*user-identifier*)[**SHARED**]**UID**(*user-identifier*)

Specifies the user identifier. The UID is a numeric value between 0 and 2 147 483 647.

When assigning a UID to a user, you should make sure that the user's default group has a GID. A user who has a UID and a current connect group that has a GID can use functions such as the TSO/E OMVS command and can access HFS files based on the UID and GID values assigned.

ADDUSER

Care should be taken in assigning 0 as the user identifier. UID 0 is considered a superuser. The superuser passes all z/OS UNIX security checks. Assigning a UID to a user ID that appears in the RACF started procedures table (ICHRIN03) should also be done with care. RACF defined started tasks that have the trusted or privileged attribute are considered superusers even if their UID is a value other than 0.

Rules:

- If the security administrator has defined the SHARED.IDS profile in the UNIXPRIV class, the UID value must be unique. Use the SHARED keyword in addition to UID to assign a value that is already in use.
- If SHARED.IDS is not defined, RACF does not require the UID to be unique. The same value can be assigned to multiple users but this is not recommended because individual user control would be lost. However, if you want a set of users to have exactly the same access to z/OS UNIX resources, you might decide to assign the same UID to more than one user.
- If the UID is not specified, the user is unable to become a z/OS UNIX user and a LISTUSER for that user ID shows NONE for the UID.

SHARED

If the security administrator has chosen to control the use of shared UIDs, this keyword must be used in addition to the UID keyword to specify the user identifier if it is already in use by at least one other user. The administrator controls shared UIDs by defining the SHARED.IDS profile in the UNIXPRIV class.

Rules:

- If the SHARED.IDS profile is not defined, SHARED is ignored.
- If SHARED is specified in the absence of UID, it is ignored.
- If the SHARED.IDS profile is defined and SHARED is specified, but the value specified with UID is not currently in use, SHARED is ignored and UNIXPRIV authority is not required.
- Field- level access checking for the UID field applies when using SHARED.
- The SHARED keyword is mutually exclusive with the AUTOUID keyword.

CPUTIMEMAX(*cpu-time*)

Specifies the RLIMIT_CPU hard limit (maximum) resource value that the user's z/OS UNIX processes receive when they are dubbed a process. The *cpu-time* you define to RACF is a numeric value between 7 and 2 147 483 647. RLIMIT_CPU indicates the *cpu-time* in seconds that a process is allowed to use. The soft limit (current) resource value is obtained from MVS. If the soft limit value from MVS is greater than the *cpu-time* value, the soft limit is used.

The value specified for CPUTIMEMAX is also used when processes are initiated by a daemon process using an exec after setuid(). In this case, both the RLIMIT_CPU hard limit and the soft limit are set to the *cpu-time* value.

For processes running in, or forked from TSO or BATCH, the *cpu-time* value has no effect. For processes created by the *rlogin* command or other daemons, the *cpu-time* is the time limit for the address space.

The value specified for CPUTIMEMAX overrides any value provided by the MAXCPUTIME parameter of BPXPRMxx. For more information, see *z/OS UNIX System Services Planning*.

FILEPROCMAX(*files-per-process*)

Specifies the maximum number of files this user is allowed to have concurrently active or open. The *files-per-process* you define to RACF is a numeric value between 3 and 262 143. FILEPROCMAX is the same as the OPEN_MAX variable defined in the POSIX standard.

FILEPROCMAX lets you limit the amount of system resources available to a user process. Select FILEPROCMAX by considering:

- For conformance to standards, set FILEPROCMAX to:
 - At least 16 to conform to the POSIX standard, and
 - At least 25 to conform to the FIPS standard.
- 256 is a commonly recommended value.
- A process can change its own value for the number of files it has active or open using the *setrlimit()* function. Only processes with appropriate privileges can increase their limits.
- The minimum value of 3 supports the standard files for a process: *stdin*, *stdout*, and *stderr*.
- The value needs to be larger than 3 to support z/OS UNIX shell users. If the value is too small, the shell might issue the message, "File descriptor not available."

The value specified for FILEPROCMAX overrides any value provided by the MAXFILEPROC parameter of BPXPRMxx. For more information, see *z/OS UNIX System Services Planning*.

HOME(*initial-directory-name*)

Specifies the user's z/OS UNIX initial directory pathname. The initial directory is part of the hierarchical file system (HFS). This is the current working directory for the user's process when the user enters the TSO/E.

When you define a HOME directory name to RACF, it can contain 1-1023 characters. The HOME pathname can consist of any characters and can be entered with or without single quotes. The following rules apply:

- If parentheses, commas, blanks, or semicolons are to be entered as part of the pathname, the character string must be enclosed in single quotes.
- If a single quote is intended to be part of the pathname, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered. The fully-qualified pathname should be specified. RACF does not ensure that a valid pathname has been specified. If you issue the ADDUSER command as a RACF operator command and you specify the pathname in lowercase, you must include the pathname within single quotations.

ADDUSER

If HOME is not specified, MVS sets the working directory for the user to “/” (the root directory). However, the default value is not placed in the user’s profile, and is not displayed when a LISTUSER command is entered.

MMAPAREAMAX(*memory-map-size*)

Specifies the maximum amount of data space storage, in pages, that can be allocated by the user for memory mappings of HFS files. Storage is not allocated until memory mappings are active. The *memory-map-size* you define to RACF is a numeric value between 1 and 16 777 216.

Use of memory map services consumes a significant amount of system memory. For each page (4KB) that is memory mapped, 96 bytes of ESQA are consumed when a file is not shared with any other users. When a file is shared by multiple users, each subsequent user after the initial user causes 32 bytes of ESQA to be consumed for each shared page. The ESQA storage is consumed when the mmap() function is invoked by the application program.

The value specified for MMAPAREAMAX overrides any value provided by the MAXMMAPAREA parameter of BPXPRMxx. For more information, see *z/OS UNIX System Services Planning*.

PROCUSERMAX(*processes-per-UID*)

Specifies the maximum number of processes this user is allowed to have active at the same time, regardless of how the process became a z/OS UNIX process. The *processes-per-UID* you define to RACF is a numeric value between 3 and 32 767. PROCUSERMAX is the same as the CHILD_MAX variable defined in the POSIX standard.

PROCUSERMAX allows you to limit user activity to optimize performance. Select PROCUSERMAX by considering:

- For conformance to standards, set PROCUSERMAX to:
 - At least 16 to conform to the POSIX standard, and
 - At least 25 to conform to the FIPS standard.
- A user with a UID of 0 is not limited by the PROCUSERMAX value because a superuser might need to be capable of logging on and using z/OS UNIX services to solve a problem.
- A low PROCUSERMAX value limits the number of concurrent processes that the user can run. A low value also limits the user’s consumption of processing time, virtual storage, and other system resources.
- Some daemons run without UID 0, and might create many address spaces. In these cases, it is necessary to set the limit high enough for the daemon associated with this user ID to run all of its processes.

Though not recommended, the same OMVS UID can be given to more than one user ID. If users share a UID, you need to define a greater number for PROCUSERMAX. An example is the user ID defined for the default OMVS segment, specified by the FACILITY class profile BPX.DEFAULT.USER.

The value specified for PROCUSERMAX overrides any value provided by the MAXPROCUSER parameter of BPXPRMxx. For more information, see *z/OS UNIX System Services Planning*.

PROGRAM(*program-name*)

Specifies the PROGRAM pathname (z/OS UNIX shell program). This is the first program started when the TSO/E command OMVS is entered or when a batch job is started using the BPXBATCH program.

When you define a PROGRAM pathname to RACF, it can contain 1-1023 characters. The PROGRAM pathname can consist of any characters and can be entered with or without single quotes. The following rules apply:

- If parentheses, commas, blanks, or semicolons are to be entered as part of the pathname, the character string must be enclosed in single quotes.
- If a single quote is intended to be part of the pathname, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered. The fully-qualified pathname should be specified. RACF does not ensure that a valid pathname has been specified. If you issue the ADDUSER command as a RACF operator command and you specify the pathname in lowercase, you must include the pathname within single quotations.

If PROGRAM is not specified or if PROGRAM is specified as blanks, MVS gives control to the default z/OS UNIX shell program. However, the default value is not placed in the user's profile, and is not displayed when a LISTUSER command is entered.

For more information about the default z/OS UNIX shell program supplied with UNIX System Services, see *z/OS UNIX System Services Planning* and *z/OS UNIX System Services User's Guide*.

THREADSMAX(*threads-per-process*)

Specifies the maximum number of pthread_create threads, including those running, queued, and exited but not detached, that this user can have concurrently active. The *threads-per-process* you define to RACF is a numeric value between 0 and 100 000. Specifying a value of 0 prevents applications run by this user from using the pthread_create service.

The value specified for THREADSMAX overrides any value provided by the MAXTHREADS parameter of BPXPRMxx. For more information, see *z/OS UNIX System Services Planning*.

OPERATIONS | NOOPERATIONS

OPERATIONS

Specifies that the new user has authorization to do maintenance operations on all RACF-protected data sets, tape volumes, and DASD volumes except those where the access list specifically limits the OPERATIONS user to a lower access authority than the operation requires.

The OPERATIONS attribute allows the user to access VM resources except those where the resource's access list specifically limits the OPERATIONS user to a lower access authority.

You establish the lower access authority for the OPERATIONS user through the PERMIT command. OPERATIONS specified on ADDUSER overrides NOOPERATIONS specified on the CONNECT command.

You must have the SPECIAL attribute to enter the OPERATIONS operand.

NOOPERATIONS

Specifies that the new user is not to have the OPERATIONS attribute. NOOPERATIONS is the default if you omit both OPERATIONS and NOOPERATIONS.

ADDUSER

OPERPARM

Specifies default information used when this user establishes an extended MCS console session.

You can control access to the entire OPERPARM segment or to individual fields within the OPERPARM segment by using field-level access checking. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

For information on planning how to use OPERPARM segments, see *z/OS MVS Planning: Operations*.

Notes:

1. You need not specify every suboperand in an OPERPARM segment. In general, if you omit a suboperand, the default is the same as the default in the CONSOLxx PARMLIB member, which can also be used to define consoles.
2. If you specify MSCOPE or ROUTCODE but do not specify a value for them, RACF uses MSCOPE(*ALL) and ROUTCODE(NONE) to update the corresponding fields in the user profile, and these values appear in listings of the OPERPARM segment of the user profile.
3. If you omit the other suboperands, RACF does not update the corresponding fields in the user's profile, and no value appears in listings of the OPERPARM segment of the profile.

ALTGRP(*alternate-console-group*)

Specifies the console group used in recovery. It can contain 1-8 characters, with valid characters being 0 through 9, A through Z, # (X'7B'), \$ (X'5B'), or @ (X'7C').

AUTH(MASTER | ALL | INFO | any others)

Specifies the authority this console has to issue operator commands.

If you omit this operand, RACF does not add this field to the user's profile. However, an extended MCS console uses AUTH(INFO) when a session is established.

MASTER

Allows this console to act as a master console, which can issue all MVS operator commands.

ALL

Allows this console to issue system control commands, input/output commands, console control commands, and informational commands.

INFO

Allows this console to issue informational commands.

CONS

Allows this console to issue console control and informational commands.

IO Allows this console to issue input/output and informational commands.

SYS

Allows this console to issue system control commands and informational commands.

AUTO(YES | NO)

Specifies whether the extended console can receive messages that have been automated by the Message Processing Facility (MPF) in the sysplex.

If you omit this operand, RACF does not add this field to the user's profile. However, an extended MCS console uses AUTO(NO) when a session is established.

CMDSYS(system-name | *)

Specifies the system to which commands issued from this console are to be sent. *System-name* must be 1-8 characters, with valid characters being A through Z, 0 through 9, and @ (X'7C'), # (X'7B'), \$ (X'5B'). If * is specified, commands are processed on the local system where the console is attached.

If you omit this operand, RACF does not add this field to the user's profile. However, an extended MCS console uses CMDSYS(*) when a session is established.

DOM(NORMAL | ALL | NONE)

Specifies whether this console receives delete operator message (DOM) requests.

If you omit this operand, RACF does not add this field to the user's profile. However, an extended MCS console uses DOM(NORMAL) when a session is established.

NORMAL

Specifies that the system queues all appropriate DOM requests to this console.

ALL

Specifies that all systems in the sysplex queue DOM requests to this console.

NONE

Specifies that no DOM requests are queued to this console.

KEY(searching-key)

Specifies a 1-8-byte character name that can be used to display information for all consoles with the specified key by using the MVS command DISPLAY CONSOLES,KEY. If specified, KEY can include A through Z, 0 through 9, # (X'7B'), \$ (X'5B'), or @ (X'7C').

If you omit this operand, RACF does not add this field to the user's profile. However, an extended MCS console uses a KEY value of NONE when a session is established.

LEVEL(message-level)

Specifies the messages that this console is to receive. Can be a list of R, I, CE, E, IN, NB or ALL. If you specify ALL, you cannot specify R, I, CE, E, or IN.

If you omit this operand, RACF does not add this field to the user's profile. However, an extended MCS console uses LEVEL(ALL) when a session is established.

NB

Specifies that the console receives *no* broadcast messages.

ALL

Specifies that the console receives all of the following messages (R, I, CE, E, IN).

R Specifies that the console receives messages requiring an operator reply.

ADDUSER

I Specifies that the console receives immediate action messages.

CE

Specifies that the console receives critical eventual action messages.

E Specifies that the console receives eventual action messages.

IN Specifies that the console receives informational messages.

LOGCMDRESP(SYSTEM | NO)

Specifies if command responses are to be logged.

If you omit this operand, RACF does not add this field to the user's profile. However, an extended MCS console uses LOGCMDRESP(SYSTEM) when a session is established.

SYSTEM

Specifies that command responses are logged in the hardcopy log.

NO

Specifies that command responses are not logged.

MFORM(*message-format*)

Specifies the format in which messages are displayed at the console. Can be a combination of J, M, S, T, and X:

J Messages are displayed with a job ID or name.

M Message text is displayed.

S Messages are displayed with the name of the originating system.

T Messages are displayed with a time stamp.

X Messages that are flagged as exempt from job name and system name formatting are ignored.

If you omit this operand, RACF does not add this field to the user's profile. However, an extended MCS console uses MFORM(M) when a session is established.

MIGID(YES | NO)

Specifies that a 1-byte migration ID is to be assigned to this console. The migration ID allows command processors that use a 1-byte console ID to direct command responses to this console.

If you omit this operand, RACF does not add this field to the user's profile. However, an extended MCS console uses MIGID(NO) when a session is established.

MONITOR(*events*)

Specifies which information should be displayed when jobs, TSO sessions, or data set status are being monitored.

If you omit this operand, RACF does not add this field to the user's profile. However, an extended MCS console uses MONITOR(JOBNAMES SESS) when a session is established. *events* can be a list of the following:

JOBNAMES | JOBNAMEST

Displays information about the start and end of each job. JOBNAMES omits the times of job start and job end. JOBNAMEST displays the times of job start and job end.

SESS I SESST

Displays information about the start and end of each TSO session. SESS omits the times of session start and session end. SESST displays them.

STATUS

Specifies that the information displayed when a data set is freed or unallocated should include the data set status.

MSCOPE(system-names I * I *ALL)

Specifies the systems from which this console can receive messages that are not directed to a specific console.

If you omit this operand, RACF does not add this field to the user's profile. However, an extended MCS console uses MSCOPE(*ALL) when a session is established.

If you specify MSCOPE but omit a value, RACF uses MSCOPE(*ALL) to update this field in the user's profile. *ALL appears in listings of the OPERPARM segment of the user's profile.

system-names

Is a list of one or more system names, where *system-name* can be any combination of A through Z, 0 through 9, # (X'7B'), \$ (X'5B'), or @ (X'7C').

* Is the system on which the console is currently active.

***ALL**

All systems.

ROUTCODE(ALL I NONE I routing-codes)

Specifies the routing codes of messages this console is to receive.

If you omit this operand, RACF does not add this field to the user's profile. However, an extended MCS console uses ROUTCODE(NONE) when a session is established.

If you specify ROUTCODE but omit a value, RACF uses ROUTCODE(NONE) to update this field in the user's profile. NONE appears in listings of the OPERPARM segment of the user's profile. The value for ROUTCODE can be one of the following:

ALL

All routing codes.

NONE

No routing codes.

routing-codes

One or more routing codes or sequences of routing codes. The routing codes can be list of *n* and *n1:n2*, where *n*, *n1*, and *n2* are integers from 1 to 128, and *n2* is greater than *n1*.

STORAGE(amount)

Specifies the amount of storage in megabytes in the TSO/E user's address space that can be used for message queuing to this console. If specified, STORAGE must be a number between 1 and 2000.

If you omit this operand, RACF does not add this field to the user's profile. However, an extended MCS console uses STORAGE(1) when a session is established and a value of 0 is listed in the OPERPARM segment of the user's profile to indicate that no storage value was specified.

ADDUSER

UD(YES | NO)

Specifies whether this console is to receive undelivered messages.

If you omit this operand, RACF does not add this field to the user's profile. However, an extended MCS console uses UD(NO) when a session is established.

OVN

Specifies OpenExtensions VM information for the user being defined. Information is stored in the OVN segment of the user's profile.

You can control access to an entire OVN segment or to individual fields within the OVN segment by using field level access checking.

FSROOT(*file-system-root*)

Specifies the pathname for the file system root.

When you define the FSROOT pathname to RACF, it can contain 1-1023 characters, consist of any character, and be entered with or without single quotes. The following rules hold:

- If parentheses, commas, blanks, or semicolons are entered as part of the pathname, the character string must be enclosed in single quotes. For example if the path name is (123), you must enter FSROOT(' (123) ').
- If a single quote is intended to be part of the pathname, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

When entering the ADDUSER command, both uppercase and lowercase are accepted and maintained in the case in which they are entered.

If you do not specify a value for FSROOT in the OVN segment, VM uses the value specified in the CP directory. If no value is specified in the CP directory, issue the OPENVM MOUNT command to mount the appropriate file system.

HOME(*initial-directory-name*)

Specifies the initial directory pathname. The initial directory is part of the file system and is the current working directory for the user's process when the user enters the OPENVM SHELL can contain 1-1023 characters, consist of any character, and be entered with or without single quotes. The following rules hold:

- If parentheses, commas, blanks, or semicolons are entered as part of the pathname, the character string must be enclosed in single quotes. For example if the path name is (123), you must enter HOME(' (123) ').
- If a single quote is intended to be part of the pathname, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

When entering the ADDUSER command, both uppercase and lowercase are accepted and maintained in the case in which they are entered.

If no value is specified for HOME in the OVN segment, VM uses the value specified in the CP directory. If no value is specified in the CP directory, VM sets the working directory for the user to "/", the root directory.

PROGRAM(*program-name*)

Specifies the PROGRAM pathname (z/OS UNIX shell program). This is the first program started when the OPENVM SHELL command is entered.

When you define a PROGRAM pathname to RACF, it can contain 1-1023 characters, consist of any character and be entered with or without single quotes. The following rules apply:

- If parentheses, commas, blanks, or semicolons are entered as part of the pathname, the character string must be enclosed in single quotes. For example if the path name is (123), you must enter PROGRAM(' (123) ').
- If a single quote is intended to be part of the pathname, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

When entering the ADDUSER command for OVM segment information, both uppercase and lowercase are accepted and maintained in the case in which they are entered. Specify the fully-qualified path name, because RACF does not ensure that a valid pathname has been specified.

If no value is specified for PROGRAM in the OVM segment, VM uses the value specified in the CP directory. If no value is specified in the CP directory, VM gives control to the default z/OS UNIX shell program (/bin/sh) when a user issues the OPENVM SHELL command.

UID(*user-identifier*)

Specifies the user identifier. The UID is a numeric value between 0 and 2 147 483 647.

Care should be taken in assigning 0 as the user identifier. UID 0 is considered a superuser.

If UID is not specified, the user is assigned the default UID of 4 294 967 295 (X'FFFFFFFF') and the LISTUSER command for that user ID shows NONE for the UID.

Note: RACF does not require the UID to be unique. You can assign the same value to multiple users, but this is not recommended because individual user control is lost. However, if you want a set of users to have exactly the same access to the OpenExtensions VM resources, you can assign the same UID to more than one user.

OWNER(*userid or group-name*)

Specifies a RACF-defined user or group to be assigned as the owner of the RACF profile for the user being added. If you omit this operand, you are defined as the owner.

PASSWORD | NOPASSWORD

PASSWORD(*password*)

Specifies the user's initial logon password. This password is always set expired, thus requiring the user to change the password at initial logon. Note that the password syntax rules your installation defines using SETROPTS PASSWORD do not apply to this password.

If you omit both PASSWORD and NOPASSWORD, or enter PASSWORD with no value, RACF takes the group name from the DFLTGRP operand as the default password.

NOPASSWORD

Specifies that the new user does not need to supply an initial logon password when first entering the system if OIDCARD is also specified. If you specify NOOIDCARD (or you allow this option to default) and you specify NOPASSWORD, you define a protected user ID that cannot be

ADDUSER

used to enter the system by any means that requires a password to be specified, such as a TSO logon, CICS signon, or batch job that specifies a password on the JOB statement. Therefore, user IDs that you assign to z/OS UNIX, UNIX daemons, started procedures, applications, servers or subsystems can be protected from being revoked when an incorrect password is entered. If the user attempts to enter the system with a password, the attempt fails. Note that the protected user ID is not revoked due to the failed password attempts even if the SETROPTS PASSWORD(REVOKE) option is in effect.

Determine which user IDs you wish to protect, ensuring that these user IDs will not be used in any circumstance where a password must be supplied. A protected user will have the PROTECTED attribute displayed in the output of the LISTUSER command. Protected users can be associated with started procedures defined in the STARTED class (preferred method) or in the started procedures table (ICHRIN03).

Notes:

1. A protected user ID can still be revoked for failed password attempts on a down-level system (OS/390 Version 2 Release 7 or earlier).
2. Starting with Release 8, protected RACF user IDs can be defined, which cannot be used for activities such as logging on to TSO or signing on to CICS. These RACF user identities allow auditing and authorization, but are not intended for users (or other systems). Kerberos information, such as a local principal name, also must not be defined for protected user IDs and these user IDs must not be used for Kerberos authentication, since Kerberos authentication failures can result in user revocation.

PROXY

Specifies information which the z/OS LDAP Server will use when acting as a proxy on behalf of a requester. The R_proxyserv (IRRSPY00) SAF callable service will attempt to retrieve this information when it is not explicitly supplied via invocation parameters. Applications or other services which use the R_proxyserv callable service, such as IBM Policy Director Authorization Services for z/OS and OS/390, may instruct their invokers to define PROXY segment information.

LDAPHOST(*ldap_url*)

Specifies the URL of the LDAP server which the z/OS LDAP Server will contact when acting as a proxy on behalf of a requester. An LDAP URL has a format such as `ldap://123.45.6:389` or `ldaps://123.45.6:636`, where `ldaps` indicates that an SSL connection is desired for a higher level of security. LDAP will also allow you to specify the host name portion of the URL using either the text form (`BIGHOST.POK.IBM.COM`) or the dotted decimal address (`123.45.6`). The port number is appended to the host name, separated by a colon ':' (`X'7A'`). See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP URLs and how to enable LDAP servers for SSL connections.

The LDAP URL that you define to RACF can consist of 10—1023 characters. A valid URL must start with either `ldap://` or `ldaps://`. RACF will allow any characters to be entered for the remaining portion of the URL, but you should ensure that the URL conforms to TCP/IP conventions. For example, parentheses, commas, blanks, semicolons, and single quotes are not typically allowed in a host name. The LDAP URL can be entered with or without single quotes, however, in both cases, it will be folded to upper case.

RACF does not ensure that a valid LDAP URL has been specified.

BINDDN(*bind_distinguished_name*)

Specifies the distinguished name (DN) which the z/OS LDAP Server will use when acting as a proxy on behalf of a requester. This DN will be used in conjunction with the BIND password if the z/OS LDAP Server needs to supply an administrator or user identity to BIND with another LDAP Server. A DN is made up of attribute value pairs, separated by commas. For example:

```
cn=Ben Gray,ou=editing,o=New York Times,c=US
cn=Lucille White,ou=editing,o=New York Times,c=US
cn=Tom Brown,ou=reporting,o=New York Times,c=US
```

See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP DNs.

When you define a BIND DN to RACF, it can contain 1—1023 characters. The BIND DN can consist of any characters and can be entered with or without single quotes. The following rules apply:

- If parentheses, commas, blanks, or semicolons are to be entered as part of the BIND DN, the character string must be enclosed in single quotes.
- If a single quote is intended to be part of the BIND DN, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered. See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP distinguished names.

If you issue the ADDUSER command as a RACF operator command and you specify the BIND DN in lowercase, you must include the BIND DN within single quotations.

RACF does not ensure that a valid BIND DN has been specified.

BINDPW

Specifies the password which the z/OS LDAP Server will use when acting as a proxy on behalf of a requester.

When you define a BIND password to RACF, it can contain 1—128 characters. The BIND password can consist of any characters (see exception below) and can be entered with or without single quotes.

Rules:

- The BIND password can not start with a left curly brace '{' (X'8B').
- If parentheses, commas, blanks, or semicolons are to be entered as part of the BIND password, the character string must be enclosed in single quotes.
- If a single quote is intended to be part of the BIND password, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered. See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP passwords.

ADDUSER

If you issue the ADDUSER command as a RACF operator command and you specify the BIND password in lowercase, you must include the BIND password within single quotations.

RACF does not ensure that a valid BIND password has been specified.

Attention:

- When the command is issued from ISPF, the TSO command buffer (including possible BINDPW password data) is written to the ISPLOG data set. As a result, you should not issue this command from ISPF or you must control the ISPLOG data set carefully.
- When the command is issued as a RACF operator command, the command and the possible BINDPW password data is written to the system log. Therefore, use of RALTER as a RACF operator command should either be controlled or you should issue the command as a TSO command.

RESTRICTED | NORESTRICTED

RESTRICTED

Specifies that global access checking is bypassed when resource access checking is performed for the new user, and neither ID(*) on the access list nor the UACC will allow access. The RESTRICTED.FILESYS.ACCESS profile in the UNIXPRIV class can also be used to bypass the z/OS UNIX 'other' permission bits during file access checking for RESTRICTED users.

Note: If your installation has profiles defined in the PROGRAM class and the user ID with the RESTRICTED attribute needs to load programs covered by one or more of these profiles, the user ID or a group to which the user is connected must be put on the access list with EXECUTE or READ authority.

NORESTRICTED

Specifies that the new user does not have the RESTRICTED attribute and access checking is performed the standard way including global access checking, ID(*), the UACC, and the z/OS UNIX 'other' permission bits as appropriate. NORESTRICTED is the default value if you omit both the RESTRICTED and NORESTRICTED keywords.

SECLABEL(*security-label*)

Specifies the user's default security label, where *security-label* is an installation-defined security label name that represents an association between a particular security level and zero or more security categories.

If the user does not enter a security label when logging on, this value becomes the user's current security label.

A security label corresponds to a particular security level (such as CONFIDENTIAL) with a set of zero or more security categories (such as PAYROLL or PERSONNEL).

When the SECLABEL class is not active, RACF ignores this operand. When no member of the SECLABEL profile exists for *security-label*, you are prompted to provide a valid *security-label*.

SECLEVEL(*security-level*)

Specifies the user's security level, where *security-level* is an installation-defined security level name that must be a member of the SECLEVEL profile in the

SECDATA class. The *security-level* that you specify corresponds to the number of the minimum security level that a user must have to access the resource.

When you specify SECLEVEL and the SECDATA class is active, RACF adds security level access checking to its other authorization checking. If global access checking does not grant access, RACF compares the security level allowed in the user profile with the security level required in the resource profile. If the security level in the user profile is less than the security level in the resource profile, RACF denies the access. If the security level in the user profile is equal to or greater than the security level in the resource profile, RACF continues with other authorization checking.

Note: RACF does not perform security level checking for a started task or user that has the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF started procedures table or STARTED class, or to other users by customer-supplied RACF exits.

When the SECDATA class is not active, RACF ignores this operand. When no member of the SECLEVEL profile exists for *security-level*, you are prompted to provide a valid *security-level*.

SPECIAL | NOSPECIAL

SPECIAL

Specifies that the new user is allowed to issue all RACF commands with all operands except the operands that require the AUDITOR attribute.

SPECIAL specified on the ADDUSER command overrides NOSPECIAL specified on the CONNECT command.

You must have the SPECIAL attribute to enter the SPECIAL operand.

NOSPECIAL

Specifies that the new user is not to have the SPECIAL attribute.

NOSPECIAL is the default if you omit both SPECIAL and NOSPECIAL.

TSO

Specifies that when you define a TSO user to RACF, you can enter any of the following suboperands to specify default TSO logon information for that user. Each suboperand defines information that RACF stores in a field within the TSO segment of the user's profile.

You can control access to an entire TSO segment or to individual fields within the TSO segment by using field-level access checking. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

ACCTNUM(account-number)

Specifies the user's default TSO account number when logging on through the TSO/E logon panel. The account number you specify must be protected by a profile in the ACCTNUM general resource class, and the user must be granted READ access to the profile. Otherwise, the user cannot log on to TSO using the specified account number.

Account numbers can consist of any characters, and can be entered with or without single quotes. The following rules apply:

- If parentheses, commas, blanks, or semicolons are to be entered as part of the account number, the character string must be enclosed in single quotes. For example, if the account number is (123), you must enter ACCTNUM(' (123) ').

ADDUSER

- If a single quote is intended to be part of the account number, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

A user can change an account number, or specify an account number if one has not been specified, using the TSO/E logon panel. RACF checks the user's authorization to the specified account number. If the user is authorized to use the account number, RACF stores the account number in the TSO segment of the user's profile, and TSO/E uses it as a default value the next time the user logs on to TSO/E. Otherwise, RACF denies the use of the account number.

Note that when you define an account number on TSO, you can specify 1-40 characters. When you define a TSO account number to RACF, you can specify only 1-39 characters.

COMMAND(*command-issued-at-logon*)

Specifies the command to be run during TSO/E logon. TSO/E uses this field to prime the COMMAND field of the logon panel. The command value can contain 1-80 characters and consist of any characters. You can enter the value with or without single quotes depending on the following rules:

- If the command value contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotes.
- If a single quote is intended to be part of the command value, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered. A user can change the command value, or specify a command if one has not been specified, using the TSO/E logon panel.

DEST(*destination-id*)

Specifies the default destination to which the system routes dynamically-allocated SYSOUT data sets. The *destination-id* must be 1-7 alphanumeric characters, beginning with an alphabetic or national character.

HOLDCLASS(*hold-class*)

Specifies the user's default hold class. The specified value must be 1 alphanumeric character, excluding national characters.

If you specify the TSO operand on the ADDUSER command but do not specify a value for HOLDCLASS, RACF uses a default value consistent with current TSO defaults.

JOBCLASS(*job-class*)

Specifies the user's default job class. The specified value must be 1 alphanumeric character, excluding national characters.

If you specify the TSO operand on the ADDUSER command but do not specify a value for JOBCLASS, RACF uses a default value consistent with current TSO defaults.

MAXSIZE(*maximum-region-size*)

Specifies the maximum region size the user can request at logon. The *maximum-region-size* is the number of 1024-byte units of virtual storage that TSO can create for the user's private address space. The specified value must be an integer in the range of 0 through 65535 if the database is shared with any MVS/370 systems, or 0 through 2096128 if the database is not shared with any MVS/370 systems.

If you specify the TSO operand on the ADDUSER command but do not specify a value for MAXSIZE, or specify MAXSIZE(0), RACF uses a default value consistent with current TSO defaults.

If values are specified for both MAXSIZE and SIZE and SIZE is greater than MAXSIZE, RACF sets SIZE equal to MAXSIZE. If a value is specified for only SIZE or MAXSIZE and SIZE is greater than MAXSIZE, the operand is ignored.

If your installation is sharing a database between MVS and VM, the administrator is allowed to specify the largest possible value. If your installation is sharing a database between MVS systems, and one of the systems sharing the database is an MVS/370 system, the administrator must not specify a value greater than 65535.

MSGCLASS(*message-class*)

Specifies the user's default message class. The specified value must be 1 alphanumeric character, excluding national characters.

If you specify the TSO operand on the ADDUSER command but do not specify a value for MSGCLASS, RACF uses a default value consistent with current TSO defaults.

PROC(*logon-procedure-name*)

Specifies the name of the user's default logon procedure when logging on through the TSO/E logon panel. The name you specify must be 1-8-alphanumeric characters and begin with an alphabetic character. The name must also be defined as a profile in the TSOPROC general resource class, and the user must be granted READ access to the profile. Otherwise, the user cannot log on to TSO using the specified logon procedure.

A user can change a logon procedure, or specify a logon procedure if one has not been specified, using the TSO/E logon panel. TSO/E checks the user's authorization to the specified logon procedure. If the user is authorized to use the logon procedure, TSO/E uses it for this session and stores the name of the procedure in the TSO segment of the user's profile for use as the default value the next time the user logs on to TSO/E. Otherwise, TSO/E denies use of the logon procedure.

SECLABEL(*security-label*)

Specifies the user's security label if one was entered on the TSO LOGON panel. On subsequent LOGONs, it appears automatically on the panel.

Note: For more information on the relationship between the TSO security label and the user's security label, see *z/OS Security Server RACF Security Administrator's Guide*.

SIZE(*default-region-size*)

Specifies the minimum region size if the user does not request a region size at logon. The default region size is the number of 1024-byte units of virtual storage available in the user's private address space at logon. The specified value must be an integer in the range of 0 through 65535 if the database is shared with any MVS/370 systems, or 0 through 2096128 if the database is not shared with any MVS/370 systems.

A user can change the minimum region size, or specify the minimum region size if one has not been specified, using the TSO/E logon panel. RACF stores this value in the TSO segment of the user's profile and TSO/E uses it as a default value the next time the user logs on to TSO/E.

ADDUSER

If values are specified for both MAXSIZE and SIZE and SIZE is greater than MAXSIZE, RACF sets SIZE equal to MAXSIZE. If a value is specified for only SIZE or MAXSIZE and SIZE is greater than MAXSIZE, the operand is ignored.

If your installation is sharing a database between MVS and VM, the administrator is allowed to specify the largest possible value. If your installation is sharing a database between MVS systems, and one of the systems sharing the database is an MVS/370 system, the administrator must not specify a value greater than 65535.

SYSOUTCLASS(*sysout-class*)

Specifies the user's default SYSOUT class. The specified value must be 1 alphanumeric character, excluding national characters.

If you specify the TSO operand on the ADDUSER command but do not specify a value for SYSOUTCLASS, RACF uses a default value consistent with current TSO defaults.

UNIT(*unit-name*)

Specifies the default name of a device or group of devices that a procedure uses for allocations. The specified value must be 1-8-alphanumeric characters.

USERDATA(*user-data*)

Specifies optional installation data defined for the user. The specified value must be 4 EBCDIC characters; valid characters are 0 through 9 and A through F.

UACC(*access-authority*)

Specifies the default value for the universal access authority for all new resources the user defines while connected to the specified default group. The universal access authorities are ALTER, CONTROL, UPDATE, READ, and NONE. (RACF does not accept EXECUTE access authority with the ADDUSER command.) If you omit this operand or specify UACC without an access authority, the default is NONE.

This operand is group-related. If a user is connected to other groups (with the CONNECT command), the user can have a different default universal access authority in each group.

WHEN([**DAYS**(*day-info*)] [**TIME**(*time-info*)])

Specifies the days of the week and the hours in the day when the user is allowed to access the system from a terminal. The day-of-week and time restrictions apply only when a user logs on to the system; that is, RACF does not force the user off the system if the end-time occurs while the user is logged on. Also, the day and time restrictions do not apply to batch jobs; the user can submit a batch job on any day and at any time.

If you omit the WHEN operand, the user can access the system at any time. If you specify the WHEN operand, you can restrict the user's access to the system to certain days of the week or to a certain time period within each day. Otherwise, you can restrict access to both certain days of the week and to a certain time period within each day.

To allow a user to access the system only on certain days, specify DAYS(*day-info*), where *day-info* can be any one of the following:

ANYDAY

The user can access the system on any day. If you omit DAYS, ANYDAY is the default.

WEEKDAYS

The user can access the system only on weekdays (Monday through Friday).

day ...

The user can access the system only on the days specified, where *day* can be MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, or SUNDAY, and you can specify the days in any order.

To allow a user to access the system only during a certain time period of each day, specify TIME(*time-info*), where *time-info* can be any one of the following:

ANYTIME

Specifies that the user can access the system at any time. If you omit TIME, ANYTIME is the default.

start-time:end-time

Specifies that the user can access the system only during the specified time period. The format of both start-time and end-time is *hhmm*, where *hh* is the hour in 24-hour notation (00 through 23) and *mm* is the minutes (00 through 59). Note that 0000 is not a valid time value.

If start-time is greater than end-time, the interval spans midnight and extends into the following day.

If you omit DAYS and specify TIME, the time restriction applies to all seven days of the week. If you specify both DAYS and TIME, the user can access the system only during the specified time period and only on the specified days.

WORKATTR

Specifies the user-specific attributes of a unit of work. z/OS elements or features such as APPC, WLM, and z/OS UNIX might use the WORKATTR segment.

WAACCN(*account-number*)

Specifies an account number for APPC/MVS processing.

You can specify a maximum of 255 EBCDIC characters.

Use the following rules when entering a value for this field:

- If the account number contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotes. For example, if the account number is (123), you must enter WAACCN(' (123) ').
- If a single quote is intended to be part of the account number, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

WAADDR_{*n*}(*address-line*)

Specifies up to four additional address lines for SYSOUT delivery. *n* can be any number from 1 to 4.

For each *address-line*, you can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase are accepted and maintained in the case in which they are entered.

Use the following rules when entering a value for this field:

- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotes. For example, if the data is (123), you must enter WAADDR(' (123) ').

ADDUSER

- If a single quote is intended to be part of the data, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

WABLDG(*building*)

Specifies the building that SYSOUT information is to be delivered to.

You can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase are accepted and maintained in the case in which they are entered.

Use the following rules when entering a value for this field:

- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotes. For example, if the data is (123), you must enter WABLDG(' (123) ').
- If a single quote is intended to be part of the data, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

WADEPT(*department*)

Specifies the department that SYSOUT information is to be delivered to.

You can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase are accepted and maintained in the case in which they are entered.

Use the following rules when entering a value for this field:

- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotes. For example, if the data is (123), you must enter WADEPT(' (123) ').
- If a single quote is intended to be part of the data, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

WANAME(*name*)

Specifies the name of the user that SYSOUT information is to be delivered to.

You can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase are accepted and maintained in the case in which they are entered.

Use the following rules when entering a value for this field:

- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotes. For example, if the data is (123), you must enter WANAME(' (123) ').
- If a single quote is intended to be part of the data, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

WAROOM(*room*)

Specifies the room that SYSOUT information is to be delivered to.

You can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase are accepted and maintained in the case in which they are entered.

Use the following rules when entering a value for this field:

- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotes. For example, if the data is (123), you must enter WAR00M(' (123) ').
- If a single quote is intended to be part of the data, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

Examples

Table 8. ADDUSER Examples

Example 1

Operation User IA0 wants to define users PAJ5 and ESH25 to RACF and assign RESEARCH as their default group.

Known User IA0 has JOIN authority to group RESEARCH and the CLAUTH attribute for the USER class.

User PAJ5 and ESH25 are not defined to RACF. User IA0 is currently connected to group RESEARCH. User IA0 wants to issue the command as a RACF TSO command.

Command ADDUSER (PAJ5 ESH25)

Defaults NAME(#####) PASSWORD(RESEARCH) OWNER(IA0) DFLTGRP(RESEARCH) AUTHORITY(USE) UACC(NONE) NOGRPACC NOADSP NOSPECIAL NOOPERATIONS NOCLAUTH NOAUDITOR NOOIDCARD

Example 2

Operation User WJE10 wants to define user RGH01 to RACF and assign PAYROLL as the default and owning group. The password is PASS, group authority is CREATE, and universal access authority is READ. User WJE10 wants to direct the command to run under the authority of user EPC at ARMNK.

Known User EPC at ARMNK has JOIN authority to group PAYROLL and the CLAUTH attribute for the USER class.

PAYROLL is not the default group of user EPC at ARMNK.

User RGH01 is not defined to RACF on node ARMNK.

The name of user RGH01 is RG Harris.

User WJE10 wants to issue the command as a RACF TSO command.

WJE10 and EPC at ARMNK have an already established user ID association.

Command ADDUSER RGH01 DFLTGRP(PAYROLL) OWNER(PAYROLL) PASSWORD(PASS) NAME('R. G. HARRIS') AUTHORITY(CREATE) UACC(READ) AT(ARMNK.EPC)

Defaults NOSPECIAL NOOPERATIONS NOCLAUTH NOOIDCARD NOAUDITOR

Example 3

Operation User RACFMIN wants to define user PIZ30 to RACF with a security category of NEWEMPLOYEE and a security level of NOSECRETS. User PIZ30 is to be allowed to use the system only on weekdays between the hours of 8:00 A.M. and 6:00 P.M.

Known User RACFMIN has the SPECIAL attribute. NEWEMPLOYEE has been defined to RACF as a valid category, and NOSECRETS has been defined as a valid security level. The new user's name is John Doe. User RACFMIN wants to issue the command as a RACF TSO command.

Command ADDUSER PIZ30 NAME('JOHN DOE') ADDCATEGORY(NEWEMPLOYEE) SECLEVEL(NOSECRETS) WHEN(DAYS(WEEKDAYS)TIME(0800:1800))

Defaults OWNER(RACFMIN) NOGRPACC NOSPECIAL NOOPERATIONS NOAUDITOR NOADSP AUTHORITY(USE)

ADDUSER

Table 8. ADDUSER Examples (continued)

Example 4	<i>Operation</i>	User TTU01 wants to define user PIZ33 to RACF. User PIZ33 will be the AUDITOR for the installation, and will have class authority to terminals and tape volumes. User PIZ33 will not be required to enter a password, but will be identified through an OIDCARD.
	<i>Known</i>	User TTU01 has the SPECIAL attribute.
		User TTU01 is connected to group RESEARCH.
		User PIZ33 is not defined to RACF.
Example 5		User TTU01 wants to issue the command as a RACF TSO command.
	<i>Command</i>	(entered in TSO foreground)
		ADDUSER PIZ33 NOPASSWORD OIDCARD CLAUTH(TAPEVOL TERMINAL) AUDITOR
		User TTU01 is prompted to enter the OIDCARD for PIZ33.
Example 6	<i>Defaults</i>	NAME(#####) OWNER(TTU01) DFLTGRP(RESEARCH) AUTHORITY(USE) UACC(NONE) NOGRPACC NOADSP NOSPECIAL NOOPERATIONS
	<i>Operation</i>	User TTU5 wants to define user RADMIN to RACF. User RADMIN will be a member of, and be owned by, the SYSINV group and have a model name of 'RADMIN.RACF.ACCESS'.
	<i>Known</i>	User TTU5 has at least JOIN authority to group SYSINV and the CLAUTH attribute for the USER class. USER TTU5 wants to issue the command as a RACF TSO command.
	<i>Command</i>	ADDUSER RADMIN DFLTGRP(SYSINV) MODEL(RACF.ACCESS) NAME('RACF ADMINISTRATOR') AUTHORITY(JOIN) ADSP UACC(NONE) OWNER(SYSINV)
Example 6	<i>Defaults</i>	NOGRPACC, NOSPECIAL, NOOPERATIONS, NOAUDITOR
	<i>Operation</i>	User KLEWIS wants to define user TBURNS to RACF and assign TSOTEST as the default group and TSOADMN as the owner of the user profile for TBURNS. The user will be allowed to use TSO and will be assigned the following TSO logon information:
		<ul style="list-style-type: none"> • Account number 98765T • Logon procedure TSPROC3 • Default job class Z • Default message class Q • Default hold class X • SYSOUT class W • Default region size of 2500 • Maximum region size of 15000.
	<i>Known</i>	<ul style="list-style-type: none"> • User KLEWIS has the SPECIAL attribute. • 98765T has been defined to RACF as a profile in the ACCTNUM general resource class, and user TBURNS has been given READ access to this profile. • TSPROC3 has been defined to RACF as a profile in the TSOPROC general resource class, and user TBURNS has been given READ access to this profile. • User TBURNS is not defined to RACF. • User TBURNS's name is T.F. Burns. • User KLEWIS wants to issue the command as a RACF TSO command.
Example 6	<i>Command</i>	ADDUSER TBURNS DFLTGRP(TSOTEST) OWNER(TSOADMN) NAME('T.F. BURNS') TSO(ACCTNUM(98765T) PROC(TSPROC3) JOBCCLASS(Z) MSGCLASS(Q) HOLDCLASS(X) SYSOUTCLASS(W) SIZE(2500) MAXSIZE(15000))
	<i>Defaults</i>	TSO(NODEST) AUTHORITY(USE) UACC(NONE) NOGRPACC NOADSP NOSPECIAL NOOPERATIONS NOCLAUTH NOAUDITOR NOOIDCARD

Table 8. ADDUSER Examples (continued)

Example 7	<p><i>Operation</i> User JSMITH wants to define user WJONES to RACF and assign SYS05 as the default group and DFPADMN as the owner of the user profile for WJONES. User WJONES is assigned the following default information to be used by DFP when the user creates a new DFP-managed data set:</p> <ul style="list-style-type: none"> • Data class DFP4DATA • Management class DFP4MGMT • Storage class DFP4STOR • Data application identifier DFP4APPL. <p><i>Known</i></p> <ul style="list-style-type: none"> • User JSMITH has the SPECIAL attribute. • DFP4MGMT has been defined to RACF as a profile in the MGMTCLAS general resource class, and user WJONES has been given READ access to this profile. • DFP4STOR has been defined to RACF as a profile in the STORCLAS general resource class, and user WJONES has been given READ access to this profile. • User WJONES is not defined to RACF. • User WJONES's name is W.E. Jones. • User JSMITH wants to issue the command as a RACF TSO command. <p><i>Command</i> ADDUSER WJONES DFLTGRP(SYS05) OWNER(DFPADMN) NAME('W.E. JONES') DFP(DATACLAS(DFP4DATA) MGMTCLAS(DFP4MGMT) STORCLAS(DFP4STOR) DATAAPPL(DFP4APPL))</p> <p><i>Defaults</i> AUTHORITY(USE) UACC(NONE) NOGRPACC NOADSP NOSPECIAL NOOPERATIONS NOCLAUTH NOAUDITOR NOOIDCARD</p>
Example 8	<p><i>Operation</i> The system administrator wants to define user DAF0 to RACF with her default group set to RESEARCH, her primary language set to American English (ENU) and her secondary language set to German (DEU).</p> <p><i>Known</i> The user's name is D. M. Brown. The profile owner is IBMUSER. The system administrator has the SPECIAL attribute. User DAF0 will have JOIN authority to group RESEARCH. The system administrator wants to issue the command as a RACF TSO command.</p> <p><i>Command</i> ADDUSER DAF0 DFLTGRP(RESEARCH) NAME('D. M. BROWN') LANGUAGE(PRIMARY(ENU) SECONDARY(DEU)) OWNER(IBMUSER) AUTHORITY(JOIN)</p> <p><i>Defaults</i> UACC(NONE) NOGRPACC NOADSP NOSPECIAL NOOPERATIONS NOCLAUTH NOAUDITOR NOOIDCARD</p>
Example 9	<p><i>Operation</i> A user with SPECIAL authority requests the addition of a new z/OS UNIX user.</p> <p><i>Known</i> The user profile will be owned by the z/OS UNIX administrator's user ID, SYSADM, and will be a member of the existing group SYSOM which is associated with a GID. The user wants to issue the command as a RACF TSO command.</p> <p><i>Command</i> ADDUSER CSMITH DFLTGRP(SYSOM) OWNER(SYSADM) NAME('C.J. SMITH') OMVS(UID(147483647) HOME(/u/CSMITH) PROGRAM(/bin/sh))</p> <p><i>Defaults</i> None</p>

ADDUSER

Table 8. ADDUSER Examples (continued)

Example 10	<i>Operation</i>	A user with SPECIAL authority requests the addition of a new DCE user.
	<i>Known</i>	The user profile is owned by the system administrator's user ID, SYSADM, and is a member of the existing group SYSOM which is associated with a GID. This DCE user has been assigned a DCE UUID of 004386ea-ebb6-1ec3-bcae-10005ac90feb and a DCE principal name of charlie. This z/OS UNIX DCE user is a principal of the /.../sivle.memphis.ibm.com DCE cell. The UUID for the /.../sivle.memphis.ibm.com DCE cell is 003456ab-ecb7-7de3-ebda-95531ed63dae.
	<i>Command</i>	<pre>ADDUSER CSMITH DFLTGRP(SYSOM) OWNER(SYSADM) NAME('C.J. SMITH') OMVS(UID(27) HOME(/u/csmith) PROGRAM(/bin/sh)) DCE(UUID(004386ea-ebb6-1ec3-bcae-10005ac90feb) + DCENAME(charlie) HOMECCELL(/.../sivle.memphis.ibm.com) + HOMEUUID(003456ab-ecb7-7de3-ebda-95531ed63dae))</pre>
Example 11	<i>Defaults</i>	DCE ([AUTOLOGIN(NO)])
	<i>Operation</i>	Lotus Notes [®] user RACFADM with SPECIAL or UPDATE authority requests the addition of a new user with Lotus Notes and NDS information.
	<i>Known</i>	The user profile is owned by RACFADM and belongs to RACFADM's current connect group SYSOM.
Example 12	<i>Command</i>	<pre>ADDUSER PCUSER1 LNOTES(SNAME('NEW-GUY 1')) NDS(UNAME(DIRADMIN))</pre>
	<i>Defaults</i>	DFLTGRP(SYSOM) OWNER(RACFADM)
	<i>Operation</i>	User RACFADM with SPECIAL or UPDATE authority requests the addition of a new z/OS UNIX user. The user specifies AUTOUID so that RACF will automatically assign an unused UID to the new user.
Example 13	<i>Known</i>	The user profile is owned by RACFADM and belongs to RACFADM's current connect group SYSOM. The BPX.NEXT.USER profile in the FACILITY class has been set up to allow automatic UID assignment.
	<i>Command</i>	<pre>ADDUSER UNIXUSR OMVS(AUTOUID HOME('/u/unixusr') CPUTIMEMAX(5000) ASSIZEMAX(40000000))</pre>
	<i>Defaults</i>	DFLTGRP(SYSOM) OWNER(RACFADM)
Example 14	<i>Operation</i>	User RACFADM with SPECIAL or UPDATE authority requests the addition of a new z/OS UNIX superuser.
	<i>Known</i>	The user profile is owned by RACFADM and belongs to RACFADM's current connect group SYSOM. Shared UIDs are being controlled, and at least one superuser already exists, so SHARED must be specified.
	<i>Command</i>	<pre>ADDUSER SUPERGUY OMVS(UID(0) SHARED HOME('/') PROGRAM('/bin/sh)) NOPASSWORD</pre>
Example 15	<i>Defaults</i>	DFLTGRP(SYSOM) OWNER(RACFADM)
	<i>Operation</i>	User RACFADM with SPECIAL authority adds the user ID PUBLIC and assigns it restricted access. User IDs RACFU00 and USER004 are added, but are not assigned any restrictions. In this example, the PUBLIC user ID does not have access to RACFU00's data sets because it has RESTRICTED access.
	<i>Known</i>	User RACFADM has SPECIAL authority.
Example 16	<i>Command</i>	<pre>ADDUSER PUBLIC RESTRICTED ADDUSER RACFU00 NORESTRICTED ADDUSER USER004 ADDSD 'RACFU00.*' UACC(READ)</pre>
	<i>Defaults</i>	USER004 has NORESTRICTED access by default.
	<i>Operation</i>	A user with SPECIAL authority requests the addition of a Security Server Network Authentication Service account within the local realm for a user whose RACF user profile is RONTOMS. MAXTKLFE is not specified, so the value specified on the definition of the local realm KERBDFLT in the REALM class is used. Note that the user's RACF password must be changed before the definition of the Network Authentication Service account is complete.
Example 17	<i>Known</i>	User RONTOMS wants to define his Security Server Network Authentication Service information.
	<i>Command</i>	<pre>ADDUSER RONTOMS KERB(KERBNAME('KerberizedUser'))</pre>
	<i>Defaults</i>	None

Table 8. ADDUSER Examples (continued)

	Example 16	<i>Operation</i>	User RACFADMIN issues a command to add a new user MRSERVER with an EIM segment and LDAP profile that is related to an LDAPBIND class for the specified user to use with EIM.
		<i>Known</i>	eimdomainALookup is a profile in the LDAPBIND class that defines the EIM LDAP values required for EIM processing
		<i>Command</i>	ADDUSER MRSERVER EIM(LDAPPROF(eimdomainALookup))
		<i>Defaults</i>	None

ALTDSD (Alter Data Set Profile)

Purpose

Use the ALTDSD command to:

- Modify an existing discrete or generic data set profile.
- Protect a single volume of either a multivolume tape data set or a multivolume, non-VSAM DASD data set. (At least one volume must already be RACF-protected.)
- Remove RACF-protection from either a single volume of a multivolume tape data set or a single volume of a multivolume, non-VSAM DASD data set. (You cannot delete the last volume from the profile.)

Changes made to discrete profiles take effect after the ALTDSD command is processed. Changes made to generic profiles do not take effect until one or more of the following steps is taken:

- The user of the data set issues the LISTDSD command:

```
LISTDSD DA(data-set-protected-by-the-profile) GENERIC
```

Note: Use the data set name, not the profile name.

- The security administrator issues the SETROPTS command:

```
SETROPTS GENERIC(DATASET) REFRESH
```

See SETROPTS command for authorization requirements.

- The user of the data set logs off and logs on again.

Note: For more information, refer to *z/OS Security Server RACF Security Administrator's Guide*.

Issuing Options

The following table identifies the eligible options for issuing the ALTDSD command:

Table 9. How the ALTDSD Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	Yes	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To create a data set profile, see “ADDSD (Add Data Set Profile)” on page 33.
- To delete a data set profile, see “DELDSD (Delete Data Set Profile)” on page 181.
- To list a data set profile, see “LISTDSD (List Data Set Profile)” on page 200.

- To permit or deny access to a data set profile, see “PERMIT (Maintain Resource Access Lists)” on page 247.
- To obtain a list of data set profiles, see “SEARCH (Search RACF Database)” on page 408.

Authorization Required

When issuing the ALTDSD command as a RACF operator command, you require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

To use the ALTDSD command, you must have sufficient authority over the profile. RACF makes the following checks until one of these conditions is met:

- You have the SPECIAL attribute.
- The data set profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You are the owner of the profile.
- The high-level qualifier of the profile name (or the qualifier supplied by the RACF naming conventions table or by a command installation exit) is your user ID.
- To assign a security label, you must have the SPECIAL attribute or have READ access to the security label profile. However, the security administrator can limit the ability to assign security labels only to users with the SPECIAL attribute.
- To access the DFP or TME segment, field-level access checking is required.

For discrete profiles only, one of the following conditions must be met:

- You are in the access list for the discrete profile and you have ALTER authority. (If you have any other level of authority, you cannot alter this profile.)
- Your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is in the access list and has ALTER authority. (If any group that RACF checked has any other level of authority, you cannot alter this profile.)
- The universal access authority is ALTER.

To use the GLOBALAUDIT operand, you must have the AUDITOR attribute, or the data set profile must be within the scope of a group in which you have the group-AUDITOR attribute.

If you have the AUDITOR attribute or the data set profile is within the scope of a group in which you have the group-AUDITOR attribute, but you do not satisfy one of the above checks, you can specify only the GLOBALAUDIT operand.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

To assign a security category to a profile, you must have the SPECIAL attribute, or the access category must be in your user profile. To assign a security level to a profile, you must have the SPECIAL attribute, or, in your own profile, a security level that is equal to, or greater than, the security level you are assigning.

ALTDSD

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the ALTDSD command is:

```
[subsystem-prefix]{ALTDSD | ALD}  
  
    (profile-name [ /password ] ...)  
    [ ADDCATEGORY(category-name ...) | DELCATEGORY [{category-name ...}*] ) ] ]  
    [ ADDVOL(volume-serial) | DELVOL(volume-serial) |  
      ALTVOL(old-volume-serial new-volume-serial) ]  
    [ AT( [ node ] .userid ...) | ONLYAT( [ node ] .userid ...) ]  
    [ AUDIT( access-attempt [ (audit-access-level) ] ...) ]  
    [ DATA('installation-defined-data') | NODATA ]  
    [ DFP(  
      RESOWNER ( userid or group-name )  
      | NORESOWNER )  
      | NODFP ]  
    [ ERASE | NOERASE ]  
    [ GENERIC | NOSET | SET ]  
    [ GLOBALAUDIT( access-attempt [(audit-access-level)] ...) ]  
    [ LEVEL(nn) ]  
    [ NOTIFY(userid) | NONOTIFY ]  
    [ OWNER(userid or group-name) ]  
    [ RETPD(nnnnn) ]  
    [ SECLABEL(seclabel-name) | NOSECLABEL ]  
    [ SECLEVEL(seclabel-name) | NOSECLEVEL ]  
    [ TME(  
      [ ROLES(role-access-specification ...) | ADDROLES(role-access-specification ...) |  
        DELROLES(role-access-specification ...) | NOROLES ]  
      )  
      | NOTME ]  
    [ UACC(access-authority) ]  
    [ UNIT(type) ]  
    [ VOLUME(volume-serial) ]  
    [ WARNING | NOWARNING ]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

profile-name

specifies the name of a discrete or generic data set profile. If you specify more than one profile name, the list of names must be enclosed in parentheses.

This operand is required and must be the first operand following ALTDSD.

Notes:

1. Because RACF uses the RACF database and not the system catalog, you cannot use alias data set names.
2. If you specify a generic profile name, RACF ignores these operands:
 - ADDVOL | DELVOL
 - ALTVOL
 - SET | NOSET
 - UNIT
 - VOLUME

/password

specifies the data set password if you are altering the profile for a password-protected data set. This operand applies only if you are using the ADDVOL and SET operands for a volume of a multivolume password-protected data set. The WRITE level password must then be specified.

If the command is executing in the foreground and you omit the password for a password-protected data set, RACF uses the logon password. You are prompted if the password you enter or the logon password is incorrect.

If the command is executing in a batch job and you either omit the password for a password-protected data set or supply an incorrect password, the operator is prompted.

You can use this operand only for tape data sets and non-VSAM DASD data sets. If you specify a generic profile, RACF ignores this operand.

ADDCATEGORY | DELCATEGORY

ADDCATEGORY(*category-name* ...)

specifies one or more names of installation-defined security categories. *category-name* must be defined as a member of the CATEGORY profile in the SECDATA class. (For information on defining security categories, see *z/OS Security Server RACF Security Administrator's Guide*.)

Specifying ADDCATEGORY on the ALTDSD command causes RACF to add any category-names you specify to any list of required categories that already exists in the data set profile. All users previously allowed to access the data set can continue to do so only if their profiles also include the additional *category-names*.

When the SECDATA class is active and you specify ADDCATEGORY, RACF performs security category checking in addition to its other authorization checking. If a user requests access to a data set, RACF compares the list of security categories in the user profile with the list of security categories in the data set profile. If RACF finds any security category in the data set profile that is not in the user's profile, RACF denies access to the data set. If the user's profile contains all the required security categories, RACF continues with other authorization checking.

Note: RACF does not perform security category checking for a started task or user that has the RACF trusted or privileged attribute. The RACF trusted or privileged attribute can be assigned to a started task through the RACF started procedures table or STARTED class, or to other users by customer-supplied RACF exits.

DELCATEGORY[(*category-name* ...!*)]

specifies one or more names of installation-defined security categories you want to delete from the data set profile. Specifying an asterisk (*) deletes all categories; RACF no longer performs security category checking for the data set profile.

Specifying DELCATEGORY by itself causes RACF to delete from the profile only undefined category names (those category names that were once known to RACF but that the installation has since deleted from the CATEGORY profile.)

ADDVOL | DELVOL

ADDVOL(*volume-serial*)

specifies that you want to RACF-protect the portion of the data set residing on this volume. At least one other portion of the data set on a different volume must already have been RACF-protected. You can use this operand only for tape data sets and non-VSAM data sets.

The DASD volume must be online unless you also specify NOSET. If it is not online and you omit NOSET, the ALTDSD command processor will, if you have TSO MOUNT authority, request that the volume be mounted.

RACF ignores this operand if you specify a generic profile name.

Note: The maximum number of volume serials for a tape data set with an entry in the TVTOC is 42.

DELVOL(*volume-serial*)

specifies that you want to remove RACF-protection from the portion of the data set residing on this volume. If no other portions of this data set on another volume are RACF-protected, the command terminates. (Use the DELDSD command to delete the profile from RACF.) You can use this operand only for tape data sets and non-VSAM DASD data sets.

The DASD volume must be online unless you also specify NOSET. If it is not online and you omit NOSET, the ALTDSD command processor requests that the volume be mounted.

RACF ignores this operand if you specify a generic profile name.

ALTVOL(*old-volume-serial new-volume-serial*)

specifies that you want to change the volume serial number in the data set profile. You can specify this operand for both VSAM and non-VSAM DASD data sets, but you cannot specify it for tape data sets. If you specify ALTVOL for a tape data set, the command fails.

When you specify ALTVOL, RACF ignores the SET and NOSET operands and modifies the data set profile, but it does not process the RACF indicator.

RACF ignores this operand if you specify a generic profile name.

To specify ALTVOL, you must have the SPECIAL attribute, or the data set profile must be within the scope of a group in which you have the group-SPECIAL attribute, or the high-level qualifier of the data set name (or the qualifier supplied by a command installation exit routine) must be your user ID.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([*node*].*userid* ...)

specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT([*node*].*userid* ...)

specifies that the command is to be directed only to the node specified by *node* where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed only to the local node.

AUDIT(*access-attempt*[*audit-access-level*] ...)*access-attempt*

specifies which new access attempts you want to log on the SMF data set. The following options are available:

ALL

indicates that you want to log both authorized accesses and detected unauthorized access attempts.

FAILURES

indicates that you want to log detected unauthorized access attempts.

NONE

indicates that you do not want any logging to be done.

SUCCESS

indicates that you want to log authorized accesses.

If you specify AUDIT without a value, RACF ignores it.

audit-access-level

specifies which access levels you want to log on the SMF data set. The levels you can specify are:

ALTER

logs ALTER access-level attempts only.

CONTROL

logs access attempts at the CONTROL and ALTER levels.

READ

logs access attempts at any level. READ is the default value if you omit audit-access-level.

UPDATE

logs access attempts at the UPDATE, CONTROL, and ALTER levels.

You cannot audit access attempts at the EXECUTE level.

DATA | NODATA**DATA('installation-defined-data')**

specifies up to 255 characters of installation-defined data to be stored in the data set profile and must be enclosed in quotes. It can also contain double-byte character set (DBCS) data.

ALTDSD

Use the LISTDSD command to list this information.

NODATA

specifies that the ALTDSD command is to delete any installation-defined data in the data set profile.

DFP | NODFP

DFP

specifies that for an SMS-managed data set, you can change the following information:

RESOWNER(*userid or group-name*) | **NORESOWNER**

specifies the user ID or group name of the actual owner of the data sets protected by the profile specified in *profile-name-1*. The name specified for RESOWNER must be a RACF-defined user or group. (The data set resource owner, or RESOWNER, is distinguished from the OWNER, which represents the user or group that owns the data set profile).

If NORESOWNER is specified, the user or group represented by the high level qualifier of the data set profile is assigned as the owner of data sets protected by the profile when SMS needs to determine the RESOWNER.

You can control access to the entire DFP segment or to individual fields within the DFP segment by using field-level access checking. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

NODFP

specifies that RACF should delete the DFP segment from the data set profile.

ERASE | NOERASE

ERASE

specifies that when SETROPTS ERASE(NOSECLEVEL) is active, data management is to physically erase the DASD data set extents at the time the data set is deleted (scratched) or released for reuse. Erasing the data set means overwriting all allocated extents with binary zeros.

This operand is ignored for the following:

- If the data set is not a DASD data set
- If SETROPTS ERASE(ALL) is specified for your installation (user and data set profile definitions are overridden)
- SETROPTS ERASE(SECLEVEL(*security_level*)) is specified for your installation (data sets equal or higher in security level are always erased, while those lower in security level are never erased)

NOERASE

specifies that data management is not to erase the DASD data set when it is deleted (scratched). If your installation has specified ERASE(ALL) on the SETROPTS command, NOERASE is meaningless. When ERASE(ALL) is in effect, data management erases all DASD data sets when they are deleted.

GENERIC | NOSET | SET

GENERIC

specifies that RACF is to treat the profile name as a generic name, even if it does not contain any generic characters.

NOSET | SET

specifies whether the data set is to be RACF-indicated. RACF ignores SET and NOSET if you do not use the ADDVOL or DELVOL operand or specify a generic profile name.

NOSET

specifies that RACF is not to change the RACF indicator for the data set.

The volume indicated in the ADDVOL or DELVOL operand does not have to be online.

To use NOSET, you must have the SPECIAL attribute, or the data set profile must be within the scope of a group in which you have the group-SPECIAL attribute, or the high-level qualifier of the data set name (or the qualifier supplied by a command installation exit) must be your user ID. If you are not authorized, RACF ignores the NOSET and ADDVOL or DELVOL operands.

SET

specifies that:

- The data set on this volume is to be RACF-indicated if you also specify the ADDVOL operand. If the indicator is already on, the command fails.
- The RACF-indicator for the data set on this volume is to be set off if you also specify the DELVOL operand. If the indicator is already off, the command fails.

For a DASD data set, the volume indicated in the ADDVOL or DELVOL operand must be online.

GLOBALAUDIT(*access-attempt*[*audit-access-level*]...)*access-attempt*

specifies which access attempts the user who has the AUDITOR attribute wants to log on the SMF data set. The following options are available:

ALL

indicates that you want to log both authorized accesses and detected unauthorized access attempts.

FAILURES

indicates that you want to log detected unauthorized access attempts.

NONE

indicates that you do not want any logging to be done.

SUCCESS

indicates that you want to log authorized accesses.

If you specify AUDIT without a value, RACF ignores it.

audit-access-level

specifies which access levels the user who has the AUDITOR attribute wants to log on the SMF data set. The levels you can specify are:

ALTER

logs ALTER access-level attempts only.

CONTROL

logs access attempts at the CONTROL and ALTER levels.

READ

logs access attempts at any level. READ is the default value if you omit audit-access-level.

UPDATE

logs access attempts at the UPDATE, CONTROL, and ALTER levels.

You cannot audit access attempts at the EXECUTE level.

To use the GLOBALAUDIT operand, you must have the AUDITOR attribute, or the profile must be within the scope of a group in which you have the group-AUDITOR attribute.

Note: Regardless of the value specified in GLOBALAUDIT, RACF always logs all access attempts specified on the AUDIT operand.

LEVEL(*nn*)

specifies a new level indicator, where *nn* is an integer between 0 and 99.

Your installation assigns the meaning of the value.

RACF includes it in all records that log data set accesses and in the LISTDSD command display.

NOTIFY | NONOTIFY

NOTIFY[(*userid*)]

specifies the user ID of a user to be notified whenever RACF uses this profile to deny access to a data set. If you specify NOTIFY without specifying a user ID, RACF takes your user ID as the default; you are notified whenever the profile denies access to a data set.

A user who is to receive NOTIFY messages should log on frequently, both to take action in response to the unauthorized access attempts the messages describe and to clear the messages from the SYS1.BROADCAST data set. (When the profile also includes WARNING, RACF might have granted access to the data set to the user identified in the message.)

Note: The user ID specified on the NOTIFY operand is not notified when the profile disallows creation or deletion of a data set. NOTIFY is only used for resource access checking, not for resource creation or deletion.

NONOTIFY

specifies that no user is to be notified when RACF uses this profile to deny access to a data set.

OWNER(*userid or group-name*)

specifies a RACF-defined user or group to be the new owner of the data set profile. If you specify a user ID as the owner of a group data set profile, the specified user must have at least USE authority in the group to which the data set profile belongs.

To change the owner of a profile, you must be the current owner of the profile or have the SPECIAL attribute, or the profile must be within the scope of a group in which you have the group-SPECIAL attribute.

Note: The user specified as the owner does not automatically have access to the data set. Use the PERMIT command to add the owner to the access list as desired.

RETPD(nnnnn)

specifies the RACF security retention period for a tape data set. The security retention period is the number of days that must elapse before a tape data set profile expires. (Note that, even though the data set profile expires, RACF-protection for data sets protected by the profile is still in effect. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.)

The number you specify must be 1 to 5 digits in the range of 0 through 65533 or, to indicate a data set that never expires, 99999.

Using RETPD to change the RACF security retention period for a data set means that the RACF security retention period and the data set retention period specified by the EXPDT/RETPD parameters on the JCL DD statement are longer be the same.

When the TAPEVOL class is active, RACF checks the RACF security retention period before it allows a data set to be overwritten. RACF adds the number of days in the retention period to the creation date for the data set. If the result is less than the current date, RACF continues to protect the data set.

When the TAPEVOL class is not active, RACF ignores the RETPD operand.

Specifying this operand for a DASD data set does not cause an error, but it has no meaning because RACF ignores the operand during authorization checking.

SECLABEL | NOSECLABEL**SECLABEL**(seclabel-name)

specifies an installation-defined security label for this profile. A security label corresponds to a particular security level (such as CONFIDENTIAL) with a set of zero or more security categories (such as PAYROLL or PERSONNEL).

RACF stores the name of the security label you specify in the data set profile if you are authorized to use that SECLABEL.

If you are not authorized to the SECLABEL or if the name you had specified is not defined as a SECLABEL profile in the SECLABEL class, the data set profile is not updated.

Note: If the SECLABEL class is active and the security label is specified in this profile, any security levels and categories in the profile are ignored.

NOSECLABEL

removes the security label, if one had been specified, from the profile.

SECLEVEL | NOSECLEVEL**SECLEVEL**(seclabel-name)

specifies the name of an installation-defined security level. This name corresponds to the number that is the minimum security level that a user must have to access the data set. The *seclabel-name* must be a member of the SECLEVEL profile in the SECADATA class.

When you specify SECLEVEL and the SECADATA class is active, RACF adds security level access checking to its other authorization checking. If global access checking does not grant access, RACF compares the security level allowed in the user profile with the security level required in the data set profile. If the security level in the user profile is less than the security level in the data set profile, RACF denies the access. If the security

ALTDSD

level in the user profile is equal to or greater than the security level in the data set profile, RACF continues with other authorization checking.

Note: RACF does not perform security level checking for a started task or user that has the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF started procedures table or STARTED class, or to other users by customer-supplied RACF exits.

If the SECDATA class is not active, RACF stores the name you specify in the data set profile. When the SECDATA class is activated and the name you specified is defined as a SECLEVEL profile, RACF can perform security level access checking for the data set profile. If the name you specify is not defined as a SECLEVEL profile and the SECDATA class is active, you are prompted to provide a valid security level name.

NOSECLEVEL

specifies that the ALTDSD command is to delete the security level name from the profile. RACF no longer performs security level access checking for the data set.

TME | NOTME

TME

specifies that information for the Tivoli Security Management Application is to be added, changed, or deleted.

Note: The TME segment fields are intended to be updated only by the Tivoli Security Management Application, which manages updates, permissions, and cross references. A security administrator should only directly update Tivoli Security Management fields on an exception basis.

ROLES(*role-access-specification ...*)

specifies a list of roles and associated access levels related to this profile.

One or more *role-access-specification* values can be specified, each separated by blanks. Each value should contain no imbedded blanks and should have the following format:

role-name:access-authority[:conditional-class:conditional-profile]

where *role-name* is a discrete general resource profile defined in the ROLE class. The *access-authority* is the authority (NONE, EXECUTE, READ, UPDATE, CONTROL, or ALTER) with which groups in the role definition should be permitted to the resource.

The *conditional-class* is a class name (APPCPORT, CONSOLE, JESINPUT, PROGRAM, TERMINAL, or SYSID) for conditional access permission, and is followed by the *conditional-profile* value, a resource profile defined in the conditional class.

ADDROLES(*role-access-specification ...*)

specifies that specific roles and access levels are to be added to the current list.

DELROLES(*role-access-specification ...*)

specifies that specific roles from the current list of roles are to be removed.

NOROLES

specifies that the entire list of roles be removed.

NOTME

specifies that RACF delete the TME segment from the profile.

UACC(*access-authority*)

specifies the universal access authority to be associated with the data sets. The universal access authorities are ALTER, CONTROL, READ, UPDATE, EXECUTE, and NONE. If you specify CONTROL for a tape data set or a non-VSAM DASD data set, RACF treats the access authority as UPDATE. If you specify EXECUTE for a tape data set or a DASD data set not used as a program library, RACF treats the access authority as NONE.

If a user accessing a data set has the RESTRICTED attribute, RACF treats the universal access authority (UACC) as NONE for that access attempt.

If you enter UACC without a value, RACF retains the old universal access authority for the data sets.

UNIT(*type*)

specifies the unit type to be added to the data set profile on which a non-VSAM data set resides. You can specify an installation-defined unit name, a generic device type, or a specific device address. RACF ignores this operand if you specify a generic profile name.

VOLUME(*volume-serial*)

specifies the volume on which the tape data set, the non-VSAM DASD data set, or the catalog for the VSAM data set resides.

If you specify VOLUME and *volume-serial* does not appear in the profile for the data set, the command fails. If you omit VOLUME and the data set name appears more than once in the RACF database, the command fails. If you omit VOLUME and the data set name appears only once in the RACF database, no volume serial checking is performed and processing continues.

RACF ignores this operand if you specify a generic profile name.

WARNING | NOWARNING**WARNING**

specifies that even if access authority is insufficient, RACF is to issue a warning message and allow access to the resource. RACF also records the access attempt in the SMF record if logging is specified in the profile.

NOWARNING

specifies that if access authority is insufficient, RACF is to deny the user access to the resource and not issue a warning message.

ALTDSD

Examples

Table 10. ALTDSD Examples

Example 1	<i>Operation</i>	User AEH0 owns data set profile PAYROLL.DEPT2.DATA and wants to assign ownership of the data set to group PAYROLL. Only users with categories of FINANCIAL and PERSONNEL and a security level of PERSONAL are to be able to access the data set.
	<i>Known</i>	Data set PAYROLL.DEPT2.DATA is RACF-defined with a discrete profile. FINANCIAL and PERSONNEL are valid categories of access; PERSONAL is a valid security level name. USER AEH0 wants to issue the command as a RACF TSO command.
	<i>Command</i>	ALTDSD 'PAYROLL.DEPT2.DATA' OWNER(PAYROLL) ADDCATEGORY(FINANCIAL PERSONNEL) SECLEVEL(PERSONAL)
	<i>Defaults</i>	None
Example 2	<i>Operation</i>	User WRH0 wants to change the universal access authority to NONE for data set RESEARCH.PROJ02.DATA and wants to have all accesses to the data set logged on SMF records. User ADMIN02 is to be notified when RACF uses this profile to deny access to the data set. The data set is to be erased when it is deleted (scratched).
	<i>Known</i>	User WRH0 has ALTER access to data set profile RESEARCH.PROJ02.DATA. User WRH0 is logged onto group RESEARCH. USER WRH0 wants to issue the command as a RACF TSO command.
		User ADMIN02 is a RACF-defined user.
		Data set RESEARCH.PROJ02.DATA is RACF-defined with a generic profile. The SETROPTS ERASE option has been specified for the installation.
Example 3	<i>Command</i>	ALTDSD 'RESEARCH.PROJ02.DATA' UACC(NONE) AUDIT(ALL(READ)) GENERIC NOTIFY(ADMIN02) ERASE
	<i>Defaults</i>	None
	<i>Operation</i>	User CD0 wants to remove RACF-protection from volume 222222 of the multivolume data set CD0.PROJ2.DATA.
	<i>Known</i>	CD0.PROJ2.DATA is a non-VSAM data set that resides on volumes 111111 and 222222 and is defined to RACF with a discrete profile. Volume 222222 is online. User CDO's TSO profile specifies PREFIX (CDO). User CD0 wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.
Example 4	<i>Command</i>	@ALTDSD PROJ2.DATA DELVOL(222222)
	<i>Default</i>	None
	<i>Operation</i>	User RVD02 wants to have all successful accesses to data set PAYROLL.ACCOUNT on volume SYS003 to be logged to the SMF data set.
	<i>Known</i>	User RVD02 has the AUDITOR attribute. User RVD02 wants to issue the command as a RACF TSO command.
Example 5	<i>Command</i>	ALTDSD 'PAYROLL.ACCOUNT' GLOBALAUDIT(SUCCESS(READ)) VOLUME(SYS003)
	<i>Defaults</i>	None
	<i>Operation</i>	User SJR1 wants to modify the installation-defined information associated with the tape data set SYSINV.ADMIN.DATA. The RACF security retention period is to be 360 days.
	<i>Known</i>	User SJR1 has ALTER authority to the data set profile. User SJR1 wants to issue the command as a RACF TSO command.
		Tape data set protection is active.
	<i>Command</i>	ALTDSD 'SYSINV.ADMIN.DATA' DATA('LIST OF REVOKED RACF USERIDS') RETPD(360)
	<i>Defaults</i>	None

Table 10. ALTDSD Examples (continued)

Example 6	<i>Operation</i>	User ADM1 wants to log all unauthorized access attempts and all successful updates to data sets protected by the generic profile SALES.ABC.*.
	<i>Known</i>	User ADM1 has the SPECIAL attribute. User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	ALTDSD 'SALES.ABC.*' AUDIT (FAILURES(READ) SUCCESS (UPDATE))
	<i>Defaults</i>	None
Example 7	<i>Operation</i>	User ADM1 owns the DFP-managed data set RESEARCH.TEST.DATA3 and wants to assign user ADM6 as the data set resource owner.
		User ADM1 wants to direct the command to run at node CLCON under the authority of user DROLLO and prohibit the command from being automatically directed to other nodes.
	<i>Known</i>	Data set RESEARCH.TEST.DATA3 is RACF-defined with a discrete profile. Users ADM1 and DROLLO at CLCON have the SPECIAL attribute, and ADM6 is defined to RACF on node CLCON. User ADM1 wants to issue the command as a RACF TSO command. Users ADM1 and DROLLO at CLCON have an already established user ID association.
	<i>Command</i>	ALTDSD 'RESEARCH.TEST.DATA3' DFP(RESOWNER(ADM6)) ONLYAT(CLCON.DROLLO)
	<i>Results</i>	The command is only processed on the node CLCON and not automatically directed to any other nodes in the RRSF configuration.

ALTGROUP (Alter Group Profile)

Purpose

Use the ALTGROUP command to change:

- The superior group of a group
- The owner of a group
- The terminal indicator for a group
- A model profile name for a group
- The installation-defined data associated with a group
- The default segment information for a group (for example, DFP or OMVS)

Issuing Options

The following table identifies the eligible options for issuing the ALTGROUP command:

Table 11. How the ALTGROUP Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	Yes	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To create a group profile, see “ADDGROUP (Add Group Profile)” on page 24.
- To delete a group profile, see “DELGROUP (Delete Group Profile)” on page 186.
- To connect a user to a group, see “CONNECT (Connect User to Group)” on page 173.
- To list information for a group profile, see “LISTGRP (List Group Profile)” on page 214.
- To remove a user from a group, see “REMOVE (Remove User from Group)” on page 376.
- To obtain a list of group profiles, see “SEARCH (Search RACF Database)” on page 408.

Authorization Required

When issuing the ALTGROUP command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

To change the superior group of a group, you must meet at least one of the following conditions:

- You must have the SPECIAL attribute

- All the following group profiles must be within the scope of a group in which you have the group-SPECIAL attribute:
 - The group whose superior group you are changing
 - The current superior group
 - The new superior group
- You must be the owner of—or have JOIN authority in—both the current and the new superior groups.

Note: You can have JOIN authority in one group and be the owner of or have the group-SPECIAL attribute in the other group.

If you have any of the following, you can specify any operand except as otherwise listed below:

- The SPECIAL attribute
- The group profile is within the scope of a group in which you have the group-SPECIAL attribute
- You are the current owner of the group.

To add, delete or alter segments such as DFP and OMVS in a group's profile:

- You must have the SPECIAL attribute
- Your installation must permit you to do so through field-level access checking.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

To specify the SHARED keyword, you must have the SPECIAL attribute or at least READ authority to the SHARED.IDS resource in the UNIXPRIV class.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the ALTGROUP command is:

ALTGROUP

```
[subsystem-prefix]{ALTGROUP | ALG}

      (group-name ...)
      [ AT([node].userid ...) | ONLYAT([node].userid ...) ]
      [ DATA('installation-defined-data') | NODATA ]
      [ DFP(
        [ DATAAPPL(application-name) | NODATAAPPL ]
        [ DATACLAS(data-class-name) | NODATACLAS ]
        [ MGMTCLAS(management-class-name)
          | NOMGMTCLAS ]
        [ STORCLAS(storage-class-name) | NOSTORCLAS ]
        )
      | NODFP ]
      [ MODEL(dsname) | NOMODEL ]
      [ OMVS(
        [ AUTOGID
          | GID ( group-identifier ) [ SHARED ]
          | NOGID ] )
      | NOOMVS ]
      [ OVM(
        [ GID(group-identifier) | NOGID ]
        )
      | NOOVM ]
      [ OWNER(userid or group-name) ]
      [ SUPGROUP(group-name) ]
      [ TERMUACC | NOTERMUACC ]
      [ TME(
        [ ROLES(profile-name ...)
          | ADDROLES(profile-name ...)
          | DELROLES(profile-name ...)
          | NOROLES ]
        )
      | NOTME ]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

group-name

Specifies the name of the group whose definition you want to change. If you specify more than one group name, the list of names must be enclosed in parentheses.

ALTGROUP

This operand is required and must be the first operand following ALTGROUP.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([*node*].*userid* ...)

Specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT([*node*].*userid* ...)

Specifies that the command is to be directed only to the node specified by *node* where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed only to the local node.

DATA | NODATA

DATA('installation-defined-data')

Specifies up to 255 characters of installation-defined data to be stored in the group profile and must be enclosed in quotes. It might also contain double-byte character set (DBCS) data.

Use the LISTGRP command to list this information.

NODATA

Specifies that the ALTGROUP command is to delete any installation-defined data in the group profile.

DFP | NODFP

DFP

Specifies that when you change the profile of a group, you can enter any of the following suboperands to add, change, or delete default values for the DFP data application, data class, management class, and storage class. DFP uses this information to determine data management and DASD storage characteristics when a user creates a new data set for a group.

DATAAPPL | NODATAAPPL

DATAAPPL(*application-name*)

Specifies the name of a DFP data application. The name you specify can contain up to 8 alphanumeric characters.

NODATAAPPL

Specifies that you want to delete the DFP data application name from the DFP segment of the group's profile.

DATACLAS | NODATACLAS

DATACLAS(*data-class-name*)

Specifies the default data class. The class name you specify can contain up to 8 alphanumeric characters.

A data class can specify some or all of the physical data set attributes associated with a new data set. During new data set allocation, data management uses the value you specify as a default unless it is preempted by a higher priority default, or overridden in some other way (for example, by JCL).

ALTGROUP

Note: The value you specify must be a valid data class name defined for use on your system. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

For information on defining DFP data classes, see *z/OS DFSMSdfp Storage Administration Reference*.

NODATACLAS

Specifies that you want to delete the default data class name from the DFP segment of the group's profile.

MGMTCLAS | NOMGMTCLAS

MGMTCLAS(*management-class-name*)

Specifies the default management class. The class name you specify can contain up to 8 alphanumeric characters.

A management class contains a collection of management policies that apply to data sets. Data management uses the value you specify as a default unless it is preempted by a higher priority default, or overridden in some other way (for example, by JCL).

Note: The value you specify must be defined as a profile in the MGMTCLAS general resource class, and the group must be granted at least READ access to the profile. Otherwise, RACF does not allow the group access to the specified MGMTCLAS. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

For information on defining DFP management classes, see *z/OS DFSMSdfp Storage Administration Reference*.

NOMGMTCLAS

Specifies that you want to delete the default management class name from the DFP segment of the group's profile.

STORCLAS | NOSTORCLAS

STORCLAS(*storage-class-name*)

Specifies the default storage class. The class name you specify can contain up to 8 alphanumeric characters.

A storage class specifies the service level (performance and availability) for data sets managed by the Storage Management Subsystem (SMS). During new data set allocation, data management uses the value you specify as a default unless it is preempted by a higher priority default, or overridden in some other way (for example, by JCL).

Note: The value you specify must be defined as a profile in the STORCLAS general resource class, and the group must be granted at least READ access to the profile. Otherwise, RACF does not allow the group access to the specified STORCLAS. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

For information on defining DFP storage classes, see *z/OS DFSMSdfp Storage Administration Reference*.

NOSTORCLAS

Specifies that you want to delete the default storage class name from the DFP segment of the group's profile.

NODFP

Specifies that RACF should delete the DFP segment from the group's profile.

MODEL | NOMODEL**MODEL(dsname)**

Specifies the name of a data set profile that RACF is to use as a model when new data set profiles are created that have *group-name* as the high-level qualifier. For this operand to be effective, the MODEL(GROUP) option on the SETROPTS command must be active. If the ALTGROUP command cannot find the *dsname* profile, it issues a warning message and places the profile name in the group entry.

RACF always prefixes *dsname* with the group name when it accesses the profile.

For information about automatic profile modeling, refer to *z/OS Security Server RACF Security Administrator's Guide*.

NOMODEL

Specifies that the ALTGROUP command is to delete the model name in the group profile.

OMVS | NOOMVS**OMVS**

Specifies z/OS UNIX System Services information for the group profile being changed.

AUTOGID | GID | NOGID

Specifies whether RACF is to automatically assign an unused GID value to the group, if a specific GID value is to be assigned or if the group identifier from the OMVS segment of the group's profile is to be deleted.

AUTOGID

Specifies that RACF is to automatically assign an unused GID value to the group. The GID value is derived from information obtained from the BPX.NEXT.USER profile in the FACILITY class. For more information on setting up BPX.NEXT.USER, see *z/OS Security Server RACF Security Administrator's Guide*.

If you are using RRSF automatic command direction for the GROUP class, the command sent to other nodes will contain an explicit assignment of the GID value which was derived by RACF on the local node.

Rules:

- AUTOGID cannot be specified if more than one group is entered.
- The AUTOGID keyword is mutually exclusive with the SHARED keyword.
- If both GID and AUTOGID are specified, AUTOGID is ignored.
- If both NOGID and AUTOGID are specified, AUTOGID is ignored.
- Field- level access checking for the GID field applies when using AUTOGID.

ALTGROUP

- AUTOGD cannot be used to reassign a GID value if one already exists for the group. If AUTOGID is specified, but the group already has a GID assigned, one of two things will happen.
 - If the pre-existing GID is unique to this group, this value will be identified in informational message IRR52177I, and the value will be left unchanged. If RACF Remote Sharing Facility (RRSF) automatic command direction is in effect for the GROUP class, then the outbound ALTGROUP command will be altered to contain the pre-existing GID value in the OMVS GID keyword.
 - If the pre-existing GID is not unique to this group, error message IRR52178I will be issued, and the command will fail. See IRR52178I for information on changing the group's existing GID value.

GID(group-identifier) [SHARED]

GID(group-identifier)

Specifies the group identifier. The GID is a numeric value between 0 and 2 147 483 647.

When a GID is assigned to a group, all users connected to that group who have a user identifier (UID) in their user profile can use functions such as the TSO/E command, OMVS, and can access the hierarchical file system (HFS) based on the GID and UID values assigned.

Notes:

1. If the security administrator has defined the SHARED.IDS profile in the UNIXPRIV class, the GID must be unique. Use the SHARED keyword in addition to GID to assign a value that is already in use.
2. If SHARED.IDS is not defined, RACF does not require the GID to be unique. The same value can be assigned to multiple groups, but this is not recommended because individual group control would be lost. However, if you want a set of groups to have exactly the same access to z/OS UNIX resources, you might decide to assign the same GID to more than one group.
3. RACF allows you to define and connect a user to more than 300 groups (which is the same as the NGROUPS_MAX variable defined in the POSIX standard), but when a process is created or z/OS UNIX group information is requested, only up to the first 300 z/OS UNIX groups are associated with the process or user.

The first 300 z/OS UNIX groups, that have GIDs, to which a user is connected are used by z/OS UNIX . LISTUSER displays the groups in the order that RACF examines them when determining which of the user's groups are z/OS UNIX groups.

See *z/OS UNIX System Services Planning* for information on NGROUPS_MAX.

SHARED

If the security administrator has chosen to control the use of shared GIDs, this keyword must be used in addition to the GID keyword to specify the group identifier if it is already in

use by at least one other group. The administrator controls shared GIDs by defining the SHARED.IDS profile in the UNIXPRIV class.

Rules:

- If the SHARED.IDS profile is not defined, SHARED is ignored.
- If SHARED is specified in the absence of GID, it is ignored.
- If the SHARED.IDS profile is defined and SHARED is specified, but the value specified with GID is not currently in use, SHARED is ignored and UNIXPRIV authority is not required.
- Field - level access checking for the GID field applies when using SHARED.
- The SHARED keyword is mutually exclusive with the AUTOGID keyword.

NOGID

Specifies that you want to delete the group identifier from the OMVS segment of the group's profile.

NOOMVS

Specifies that RACF delete the OMVS segment from the group's profile.

OVM | NOOVM

OVM

Specifies OpenExtensions VM information for the group profile being changed.

GID | NOGID

GID(*group-identifier*)

Specifies the group identifier. The GID is a numeric value between 0 and 2 147 483 647.

Notes:

1. RACF does not require the GID to be unique. The same value can be assigned to multiple groups, but this is not recommended because individual group control would be lost. However, if you want a set of groups to have exactly the same access to the OpenExtensions VM resources, you might decide to assign the same GID to more than one group.
2. Exercise caution when changing the GID for a group. The following situations might occur:
 - If the file system contains files that contain the old GID as the file owner GID, the members of the group lose access to those files, depending on the permission bits associated with the file.
 - If files exist with an owner GID equal to the group's new GID value, the members of the group gain access to these files.
 - If another group is subsequently added with the old value as its GID, the members of the group might have access to the old files.

ALTGROUP

- If you have an EXEC.Ggid profile in the VMPOSIX class for the old GID value, make sure you delete this profile and create another to reflect the new value.
3. The value defined for the NGROUPS_MAX variable in the ICHNGMAX macro on VM defines the maximum number of OpenExtensions VM groups to be associated with an OpenExtensions VM process or user. The NGROUPS_MAX variable on VM is a number between 32 and 125, inclusive. However, RACF allows you to define and connect a user to more than the number of groups defined in this variable. If the NGROUPS_MAX variable is n and a process is created or OpenExtensions VM group information is requested, only up to the first n OpenExtensions VM groups are associated with the process or user. The first n OpenExtensions VM groups to which a user is connected are used by OpenExtensions VM. LISTUSER displays the groups in the order that RACF examines them when determining which of the user's groups are OpenExtensions VM groups.
See z/OS Security Server RACF Macros and Interfaces for information on NGROUPS_MAX.

NOGID

Specifies that you want to delete the group identifier from the OVM segment of the group's profile.

If NOGID is specified for the group, the default GID of 4 294 967 295 (X'FFFFFFFF') is assigned on VM. The LISTGRP command displays the field name followed by the word "NONE".

NOOVM

Specifies that RACF delete the OVM segment from the group's profile.

OWNER(userid or group-name)

Specifies a RACF-defined user or group you want to be the new owner of the group.

To change the owner of a group, you must be the current owner of the group, or have the SPECIAL attribute, or have the group-SPECIAL attribute in the group owning the profile.

If you specify a group name, then OWNER and SUPGROUP must specify the same group name.

SUPGROUP(group-name)

Specifies the name of the RACF-defined group you want to make the new superior group for the group profile you are changing.

The new superior group must not be the same as the current one, and it must not have any level of subgroup relationship to the group you are changing.

To change a superior group, you must have the SPECIAL attribute, the group profile must be within the scope of a group in which you have the group-SPECIAL attribute, or you must have JOIN authority in, or be the owner of, both the current and new superior groups. Note that you can have JOIN authority in one group and be the owner of or have the group-SPECIAL attribute in the other group.

If owner is a group name, OWNER and SUPGROUP must specify the same group name.

TERMUACC | NOTERMUACC

TERMUACC

Specifies that during terminal authorization checking, RACF is to allow the use of the universal access authority for a terminal when it checks whether a user in the group is authorized to access a terminal.

NOTERMUACC

Specifies that the group or a user connected to the group must be authorized (using the PERMIT command with at least READ authority) to access a terminal.

TME | NOTME

TME

Specifies that information for the Tivoli Security Management Application is to be added, changed, or deleted.

Note: The TME segment fields are intended to be updated only by the Tivoli Security Management Application, which manages updates, permissions, and cross references. A security administrator should only directly update Tivoli Security Management fields on an exception basis.

ROLES | NOROLES | ADDROLES | DELROLES

ROLES(*profile-name*)

Specifies a list of roles that reference this group.

Profile-name should be the name of a defined role, which is a discrete general resource profile in the ROLE class.

ADDROLES(*profile-name*)

Specifies a list of roles that reference this group.

Profile-name should be the name of a defined role, which is a discrete general resource profile in the ROLE class.

DELROLES(*profile-name*)

Specifies that specific roles from the current list of roles are to be removed.

Profile-name should be the name of a defined role, which is a discrete general resource profile in the ROLE class.

NOROLES

Specifies that the entire list of roles be removed.

NOTME

Specifies that RACF delete the TME segment from the group profile.

Examples

Table 12. ALTGROUP Examples

Example 1

Operation User WJB10 wants to change the superior group and owning group for PROJECTA from RESEARCH to PAYROLL. Users connected to group PROJECTA are authorized access to terminals according to the universal access authority of the terminal.

Known User WJB10 has JOIN authority in RESEARCH and is the owner of PAYROLL.

PROJECTA is a subgroup of RESEARCH.

User WJB10 wants to issue the command as a RACF TSO command.

Command ALTGROUP PROJECTA SUPGROUP(PAYROLL) OWNER(PAYROLL) TERMUACC

Defaults None

ALTGROUP

Table 12. ALTGROUP Examples (continued)

Example 2	<i>Operation</i>	User MULES wants to change the superior group for PROJECTB from SYS1 to RESEARCH and assign RESEARCH as the new owner.
	<i>Known</i>	User MULES has the SPECIAL attribute.
		PROJECTB is a subgroup of SYS1. User MULES wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.
	<i>Command</i>	@ALTGROUP PROJECTB SUPGROUP(RESEARCH) OWNER(RESEARCH)
	<i>Defaults</i>	None
Example 3	<i>Operation</i>	User SJR2 wants to change the installation-defined information associated with the RSC1 group and delete the model name. User SJR2 wants to direct the command to run under the authority of user ANW01.
	<i>Known</i>	User SJR2 is the owner of group RSC1. User SJR2 wants to issue the command as a RACF TSO command. SJR2 and ANW01 have an already established user ID association. User ANW01 is the owner of group RSC1.
	<i>Command</i>	ALTGROUP RSC1 DATA('RESOURCE USAGE ADMINISTRATION') NOMODEL AT(.ANW01)
	<i>Defaults</i>	Command direction defaults to the local node.
Example 4	<i>Operation</i>	User BILLC wants to make the following changes to the profile for group PROJECT6:
		<ul style="list-style-type: none">• Change the default DFP management class to MCLASS7• Change the default DFP storage class to SCLASS3• Change the default DFP data class to DCLASS15• Delete the default DFP data application.
	<i>Known</i>	<ul style="list-style-type: none">• User BILLC has the SPECIAL attribute.• Group PROJECT6 has been defined to RACF, and PROJECT6's group profile contains a DFP segment.• MCLASS7 has been defined to RACF as a profile in the MGMTCLAS general resource class, and group PROJECT6 has been given READ access to this profile.• SCLASS3 has been defined to RACF as a profile in the STORCLAS general resource class, and group PROJECT6 has been given READ access to this profile.• User BILLC wants to issue the command as a RACF TSO command.
	<i>Command</i>	ALTGROUP PROJECT6 DFP(MGMTCLAS(MCLASS7) STORCLAS(SCLASS3) DATACLAS(DCLASS15) NODATAAPPL))
	<i>Defaults</i>	None

ALTUSER (Alter User Profile)

Purpose

Use the ALTUSER command to change the information in a user's profile, including the user's system-wide attributes and authorities. The user profile consists of a RACF segment and, optionally, other segments such as a TSO segment or a DFP segment. You can use this command to change information in any segment of the user's profile.

When you change a user's level of authority in a group (using the **AUTHORITY** operand), RACF updates the appropriate group profile. When you change a user's default universal access authority for a group (using the **UACC** operand), RACF changes the appropriate connect profile. For all other changes, RACF changes the user's profile.

Note: If the user is currently logged on, changes to the attributes (except for **OWNER** and **AUTHORITY**) do not take effect until the next time the user logs on, even though the **LISTUSER** command shows the new values.

Attention:

- When the ALTUSER command is issued from ISPF, the TSO command buffer (including password data) is written to the ISPLOG data set. As a result, you should not issue this command from ISPF or you must control the ISPLOG data set carefully.
- If the ALTUSER command is issued as a RACF operator command, the command and all data (including password data) is written to the system log. Therefore, use of ALTUSER as a RACF operator command should either be controlled or you should issue the command as a TSO command.

Note that you cannot:

- Use the ALTUSER command to change a user ID association; you must use the **RACLINK** command.
- Use the ALTUSER command for profiles in the **DIGTCERT** class.
- Use the ALTUSER command for user IDs that have mixed-case characters, such as **irrcerta**, **irrsitec**, and **irrmulti** (which are associated with digital certificates).

Issuing Options

The following table identifies the eligible options for issuing the ALTUSER command:

Table 13. How the ALTUSER Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	Yes	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, "RACF TSO commands" on page 15.

ALTUSER

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To add a user profile, see “ADDUSER (Add User Profile)” on page 48.
- To delete a user profile, see “DELUSER (Delete User Profile)” on page 189.
- To display information from a user profile, see “LISTUSER (List User Profile)” on page 223.
- To administer user ID associations, see “RACLINK (Administer User ID Associations)” on page 296.
- To obtain a list of user profiles, see “SEARCH (Search RACF Database)” on page 408.

Authorization Required

When issuing the ALTUSER command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

The level of authority required depends on which of the user's attributes you want to change.

- If you have the SPECIAL attribute, you can use all the operands except UAUDIT/NOUAUDIT.
- To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).
- To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.
- If the owner of the user profile is within the scope of a group in which you have the group-SPECIAL attribute, you can use all of the operands except SPECIAL, AUDITOR, OPERATIONS, NOEXPIRED and UAUDIT/NOUAUDIT.
- If you are the owner of the user's profile, you can use any of the following operands for user-related attributes:

ADSP NOADSP	MODEL NOMODEL	PASSWORD NOPASSWORD
DATA NODATA	NAME	RESTRICTED NORESTRICTED
DFLTGRP	OIDCARD NOOIDCARD	RESUME
GRPACC NOGRPACC	OWNER	REVOKE
WHEN		

- Each user can change his or her name field or default group (NAME and DFLTGRP operands). If an installation specifies MODEL(USER) on the SETROPTS command, each user can also change his or her model data set profile name (using the MODEL operand).
- You can use the ADSP | NOADSP, RESTRICTED | NORESTRICTED, GROUP, AUTHORITY, and UACC operands for group-related user attributes if you have JOIN or CONNECT authority, or if the group profile is within the scope of a group in which you have the group-SPECIAL attribute, or if you are the owner of the specified group.

- To specify the AUDITOR/NOAUDITOR, SPECIAL/NOSPECIAL, and OPERATIONS/NOOPERATIONS operands as system-wide user attributes, you must have the SPECIAL attribute.
- To specify the UAUDIT/NOUAUDIT operand, either you must have the AUDITOR attribute, or the user profile must be within the scope of a group in which you have the group-AUDITOR attribute.
- You can specify the CLAUTH and NOCLAUTH operands if you are the owner of the user's profile and have the CLAUTH attribute for the class to be added or deleted.
- To assign a security category to a profile, or to delete a category from a profile, you must have the SPECIAL attribute, or the category must be in your user profile.
- To assign a security level to a profile, or to delete a security level from a profile, you must have the SPECIAL attribute, or, in your own profile, a security level that is equal to or greater than the security level you are assigning or deleting.
- To assign a security label to a profile, or to delete a security label from a profile, you must have the SPECIAL attribute, or, in your own profile, a security label that is equal to or greater than the security label you are assigning or deleting. However, the security administrator can limit the ability to assign or delete security labels to only users with the SPECIAL attribute.
- To change information within a segment other than the base segment, you must have one of the following:
 - The SPECIAL attribute
 - At least UPDATE authority to the desired field within the segment through field-level access control

For information on field-level access checking, see *z/OS Security Server RACF Security Administrator's Guide*.

- To reset passwords and resume user IDs, you need access to the IRR.PASSWORD.RESET resource in the FACILITY class.
 - If you have READ access you can:
 - Use the PASSWORD operand, provided the user being altered does not have the SPECIAL, OPERATIONS, or AUDITOR attribute.
 - Use the RESUME operand, without specifying a date, provided the user being altered does not have the SPECIAL, OPERATIONS, or AUDITOR attribute.
 - If you have UPDATE access you can:
 - Use the NOEXPIRED operand (in conjunction with the PASSWORD operand) provided the user being altered does not have the SPECIAL, OPERATIONS, or AUDITOR attribute.
- Failed access attempts to IRR.PASSWORD.RESET are not logged. Rather, these attempts are logged as ALTUSER command violations. Successful accesses to IRR.PASSWORD.RESET are logged at the installation's discretion. For more information on logging, see *z/OS Security Server RACF Auditor's Guide*.
- To specify the SHARED keyword, you must have the SPECIAL attribute or at least READ authority to the SHARED.IDS resource in the UNIXPRIV class.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the ALTUSER command is:

ALTUSER

```
[subsystem-prefix]{ALTUSER | ALU}

    (userid ...)
    [ ADDCATEGORY(category-name ...)
    | DELCATEGORY [ (category-name ...!*) ]
    [ ADSP | NOADSP ]
    [ AT([node].userid ...)
    | ONLYAT([node].userid ...) ]
    [ AUDITOR | NOAUDITOR ]
    [ AUTHORITY(group-authority) ]
    [ CICS(
        [ OPCODE(operator-class ...)
          | ADDOPCLASS(operator-class ...)
          | DELOPCLASS(operator-class ...)
          | NOOPCLASS ]
        [ OPIDENT(operator-id) | NOOPIDENT ]
        [ OPPRTY(operator-priority) | NOOPRTY ]
        [ TIMEOUT(timeout-value) | NOTIMEOUT ]
        [ XRFSSOFF(FORCE | NOFORCE) | NOXRFSSOFF ]
        )
    | NOCICS ]
    [ {CLAUTH | NOCLAUTH} (class-name ...) ]
    [ DATA('installation-defined-data') | NODATA ]
    [ DCE( [ AUTOLOGIN(YES|NO) | NOAUTOLOGIN ]
        [ DCENAME(user-principal-name) | NODCENAME ]
        [ HOMECCELL(dce-cell-name) | NOHOMECCELL ]
        [ HOMEUUID(home-cell-universal-unique-identifier)
          | NOHOMEUUID ]
        [ UUID(universal-unique-identifier) | NOUUID ]
        )
    | NODCE ]
    [ DFLTGRP(group-name) ]
    [ DFP(
        [ DATAAPPL(application-name) | NODATAAPPL ]
        [ DATACLAS(data-class-name) | NODATACLAS ]
        [ MGMTCLAS(management-class-name)
          | NOMGMTCLAS ]
        [ STORCLAS(storage-class-name) | NOSTORCLAS ]
        )
    | NODFP ]
    [ EIM (LDAPPROF(ldapbind_profile) | NOLDAPPROF)
    | NOEIM ]
    [ EXPIRED | NOEXPIRED ]
    [ GROUP(group-name) ]
    [ GRPACC | NOGRPACC ]
    [ KERB (
        [ ENCRYPT (
            [ DES | NODES ]
            [ DES3 | NODES3 ]
            [ DESD | NODESD ] )
        | NOENCRYPT ]
        [ KERBNAME(kerberos-principal-name) | NOKERBNAME ]
        [ MAXTKTLFE(max-ticket-life) | NOMAXTKTLFE ] )
    | NOKERB ]
    [ LANGUAGE(
        [ PRIMARY(language) | NOPRIMARY ]
        [ SECONDARY(language) | NOSECONDARY ]
        )
    | NOLANGUAGE ]
```

```

[ LNOTES( [ SNAME(short-name) | NOSNAME ] )
| NOLNOTES ]
[ MODEL(dsname) | NOMODEL ]
[ NAME(user-name) ]
[ NDS( [ UNAME(user-name) | NOUNAME ] )
| NONDS ]
[ NETVIEW(
  [ CONSNAME(console-name | NOCONSNAME ]
  [ CTL(GENERALIGLOBALISPECIFIC) | NOCTL ]
  [ DOMAINS(domain-name ...)
    | ADDDOMAINS(domain-name...)
    | DELDOMAINS(domain-name...)
    | NODOMAINS ]
  [ IC('command' | command-list)
    | NOIC ]
  [ MSGRECV(R(NOIYES) | NOMSGRECV(R]
  [ NGMFADMN(NOIYES) | NONGMFADMN ]
  [ NGMFVSPN(view-span) | NONGMFVSPN ]
  [ OPCLASS(class ...)
    | ADDOPCLASS(class ...)
    | DELOPCLASS(class ...)
    | NOOPCLASS ] )
| NONETVIEW ]
[ OIDCARD | NOOIDCARD ]
[ OMVS(
  [ ASSIZEMAX(address-space-size) | NOASSIZEMAX ]
  [ AUTOUID | UID ( user-identifier ) [ SHARED ] | NOUID ]
  [ CPUTIMEMAX(cpu-time) | NOCPUTIMEMAX ]
  [ FILEPROCMA(X(files-per-process)
    | NOFILEPROCMA(X]
  [ HOME(directory-pathname) | NOHOME ]
  [ MMAPAREAMAX(memory-map-size)
    | NOMMAPAREAMAX ]
  [ PROCUSERMAX(processes-per-UID)
    | NOPROCUSERMAX ]
  [ PROGRAM(program-name) | NOPROGRAM ]
  [ THREADSMAX(threads-per-process)
    | NOTHEADSMAX ] )
| NOOMVS ]
[ OPERATIONS | NOOPERATIONS ]

```

ALTUSER

```

[ OPERPARM(
  [ ALTGRP(alternate-console-group) | NOALTGRP ]
  [ AUTH(operator-authority) | NOAUTH ]
  [ AUTO( YES | NO ) | NOAUTO ]
  [ CMDSYS(system-name) | NOCMDSYS ]
  [ DOM( NORMAL | ALL | NONE ) | NODOM ]
  [ KEY(searching-key) | NOKEY ]
  [ LEVEL(message-level) | NOLEVEL ]
  [ LOGCMDRESP( SYSTEM | NO ) | NOLOGCMDRESP ]
  [ MFORM(message-format) | NOMFORM ]
  [ MIGID( YES | NO ) | NOMIGID ]
  [ MONITOR(event) | NOMONITOR ]
  [ MSCOPE( system-name... | * | *ALL )
  | ADDMSCOPE(system-name...)
  | DELMSCOPE(system-name...) | NOMSCOPE ]
  [ ROUTCODE( ALL | NONE | routing-codes )
  | NOROUTCODE ]
  [ STORAGE(amount) | NOSTORAGE ]
  [ UD( YES | NO ) | NOUD ]
)
| NOOPERPARM ]
[ OVM( [ FSROOT(file-system-root) | NOFSROOT ]
  [ HOME(initial-directory-name) | NOHOME ]
  [ PROGRAM(program-name) | NOPROGRAM ]
  [ UID (user-identifier) | NOUID ] )
| NOOVM ]
[ OWNER(userid or group-name) ]
[ PASSWORD (password) | NOPASSWORD ]
[ PROXY [ (
  [ LDAPHOST (ldap_url) | NOLDAPHOST ]
  [ BINDDN(bind_distinguished_name) | NOBINDDN ]
  [ BINDPW(bind_password) | NOBINDPW ]
  | NOPROXY ]
[ RESTRICTED | NORESTRICTED ]
[ RESUME [(date)] ]
[ REVOKE [(date)] ]
[ SECLABEL(seclabel-name) | NOSECLABEL ]
[ SECLEVEL(seclabel-name) | NOSECLEVEL ]
[ SPECIAL | NOSPECIAL ]
[ TSO (
  [ ACCTNUM(account-number) | NOACCTNUM ]
  [ COMMAND(cmd-issued-at-logon) | NOCOMMAND ]
  [ DEST(destination-id) | NODEST ]
  [ HOLDCLASS(hold-class) | NOHOLDCLASS ]
  [ JOBCLASS(job-class) | NOJOBCLASS ]
  [ MAXSIZE(maximum-region-size) | NOMAXSIZE ]
  [ MSGCLASS(message-class) | NOMSGCLASS ]
  [ PROC(logon-procedure-name) | NOPROC ]
  [ SECLABEL(seclabel-name) | NOSECLABEL ]
  [ SIZE(default-region-size) | NOSIZE ]
  [ SYSOUTCLASS(sysout-class) | NOSYSOUTCLASS ]
  [ UNIT(unit-name) | NOUNIT ]
  [ USERDATA(user-data) | NOUSERDATA ] )
| NOTSO ]
[ UACC(access-authority) ]
[ UAUDIT | NOAUDIT ]
[ WHEN( [DAYS(day-info)] [TIME(time-info)] ) ]

```

```

[ WORKATTR(
  [ WAACNT(account-number) | NOWAACNT ]
  [ WAADDR1(address-line-1) | NOWAADDR1 ]
  [ WAADDR2(address-line-2) | NOWAADDR2 ]
  [ WAADDR3(address-line-3) | NOWAADDR3 ]
  [ WAADDR4(address-line-4) | NOWAADDR4 ]
  [ WABLDG(building) | NOWABLDG ]
  [ WADEPT(department) | NOWADEPT ]
  [ WANAME(name) | NOWANAME ]
  [ WAROOM(room) | NOWAROOM ] )
| NOWORKATTR ]

```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

userid

Specifies the RACF-defined user or users whose profile you want to change. If you specify more than one user ID, the list must be enclosed in parentheses.

This operand is required and must be the first operand following ALTUSER.

ADDCATEGORY | DELCATEGORY

ADDCATEGORY(*category-name*)

Specifies one or more names of installation-defined security categories. The names you specify must be defined as members of the CATEGORY profile in the SECDATA class. For information on defining security categories, see *z/OS Security Server RACF Security Administrator's Guide*.

When the SECDATA class is active and you specify ADDCATEGORY, RACF performs security category checking in addition to its other authorization checking. If a user requests access to a data set, RACF compares the list of security categories in the user profile with the list of security categories in the data set profile. If RACF finds any security category in the data set profile that is not in the user's profile, RACF denies access to the data set. If the user's profile contains all the required security categories, RACF continues with other authorization checking.

Note: RACF does not perform security category checking for a started task or user that has the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task

ALTUSER

through the RACF started procedures table or STARTED class, or to other users by customer-supplied RACF exits.

DELCATEGORY[(*category-name*...!*)]

Specifies one or more names of the installation-defined security categories you want to delete from the user profile. Specifying an asterisk (*) deletes all categories; the user no longer has access to any resources protected by security category checking.

Specifying DELCATEGORY without *category-name* causes RACF to delete only undefined category names (those names that once were valid names but that the installation has since deleted from the CATEGORY profile).

ADSP | NOADSP

ADSP

Assigns the ADSP attribute to the user. This means that all permanent tape and DASD data sets the user creates are automatically RACF-protected by discrete profiles. ADSP specified on the ALTUSER command overrides NOADSP specified on the CONNECT command.

The ADSP attribute has no effect (even if assigned to a user) if SETROPTS NOADSP is in effect.

NOADSP

specifies that the user no longer has the ADSP attribute.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT[(*node*).*userid* ...]

Specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT[(*node*).*userid* ...]

Specifies that the command is to be directed only to the node specified by *node* where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed only to the local node.

AUDITOR | NOAUDITOR

AUDITOR

Specifies that the user is to have full responsibility for auditing the use of system resources. An AUDITOR user can control the logging of detected accesses to any RACF-protected resources during RACF authorization checking and accesses to the RACF database.

You must have the SPECIAL attribute to enter the AUDITOR operand.

NOAUDITOR

Specifies that the user no longer has the AUDITOR attribute.

You must have the SPECIAL attribute to enter the NOAUDITOR operand.

AUTHORITY(*group-authority*)

Specifies the new level of authority the user is to have in the group specified in the GROUP operand. The valid group authority values are USE, CREATE, CONNECT, and JOIN as described in "Group authorities" on page 13. If you

specify **AUTHORITY** without *group-authority*, RACF ignores the operand and the existing group authority remains unchanged.

CICS | NOCICS

Adds, alters, or deletes CICS operator information for a CICS terminal user. This operand requires CICS/ESA 3.2.1 or later.

If you are adding a CICS segment to a user profile, omitting a suboperand is equivalent to omitting the suboperand on the **ADDUSER** command. If you are changing an existing CICS segment in a user profile, omitting a suboperand leaves the existing value for that suboperand unchanged.

You can control access to the entire CICS segment or to individual fields within the CICS segment by using field-level access checking. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

OPCLASS | ADDOPCLASS | DELOPCLASS | NOOPCLASS

Where *operator-class1*, *operator-class2* are numbers in the range of 1 through 24, defined as two digits. These numbers represent classes assigned to this operator to which BMS (basic mapping support) messages are routed.

OPCLASS(*operator-class ...*)

Specifies the list of classes assigned to this operator to which BMS messages are routed.

ADDOPCLASS(*operator-class ...*)

Adds to the list of classes assigned to this operator to which BMS messages are routed.

DELOPCLASS(*operator-class ...*)

Deletes only the specified classes from the list of classes assigned to this operator to which BMS messages are routed.

NOOPCLASS

Deletes all operator classes from this profile and returns the user to the CICS defaults for this field. This field no longer appears in **LISTUSER** output.

OPIDENT | NOOPIDENT

OPIDENT(*operator-id*)

Specifies a 1-3 character identification of the operator for use by BMS.

Operator identifiers can consist of any characters, and can be entered with or without single quotes. The following rules apply:

- If parentheses, commas, blanks, or semicolons are to be entered as part of the operator identifier, the character string must be enclosed in single quotes. For example, if the operator identifier is (1), you must enter **OPIDENT(' (1) ')**.
- If a single quote is intended to be part of the operator identifier, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

NOOPIDENT

Deletes the operator identification and returns the user to the CICS default for this field. The **OPIDENT** field defaults to blanks in the RACF user profile, and blanks appear for the field in **LISTUSER** output.

OPPRTY(*operator-priority*) | NOOPPRTY

Specifies a number in the range of 0 through 255 that represents the priority of the operator.

ALTUSER

NOOPRTY deletes the operator priority and returns the user to the CICS default for this field.

This field defaults to zeros in the RACF user profile, and zeros appear for the field in LISTUSER output.

TIMEOUT(*timeout-value*) | NOTIMEOUT

Specifies the time, in hours and minutes, that the operator is allowed to be idle before being signed off. The value for TIMEOUT can be entered in the form *m*, *mm*, *hmm*, or *hhmm*, where the value for *m* or *mm* can range from 00 to 59, or 00 to 60 if *h* or *hh* is not specified or specified as 0 or 00. The value for *h* or *hh* can range from 00 to 99.

Note: If your installation has multiple levels of CICS sharing the RACF database, be aware that versions of CICS prior to CICS 4.1 allow TIMEOUT values only in the form *mm*, where *mm* is 0 to 60. A specification of 0200 is interpreted as two hours on a CICS 4.1 system, but causes an earlier CICS system to see a timeout value of 0 and assume no timeout is to be used. To avoid this problem, users with mixed CICS systems might wish to specify a timeout value in the form *hhmm* and ensure the *mm* value is not zero. For example, you could specify a TIMEOUT value such as 0159, which is interpreted as 1 hour and 59 minutes on a CICS 4.1 system, and is interpreted as 59 minutes on an earlier CICS system.

If this suboperand is omitted, there is no change to this field.

NOTIMEOUT Deletes the timeout value and returns the user to the CICS default for this field.

This field defaults to zeros in the RACF user profile, and zeros appear for the field in LISTUSER output.

XRFSOFF(FORCE | NOFORCE) | NOXRFSOFF

Specifies that the user is to be signed off by CICS when an XRF takeover occurs.

NOXRFSOFF returns the user to the CICS default for this field.

This field defaults to NOFORCE in the RACF user profile, and NOFORCE appears in LISTUSER output.

NOCICS

Deletes the CICS segment from a user profile. No CICS information appears in LISTUSER output.

CLAUTH | NOCLAUTH

CLAUTH(*class-name* ...)

Specifies the classes in which the user is allowed to define profiles to RACF for protection, in addition to the classes previously allowed for the user. Classes you can specify are USER, and any resource class defined in the class descriptor table. RACF adds the class names you specify to the class names previously specified for this user.

To enter the CLAUTH operand, you must have the SPECIAL attribute, or the user's profile must be within the scope of a group in which you have the group-SPECIAL attribute and have the CLAUTH attribute, or you must be the owner of the user's profile and have the CLAUTH attribute for the class to be added.

Note: The CLAUTH attribute has no meaning for the FILE and DIRECTORY classes.

NOCLAUTH(*class-name ...*)

Specifies that the user is not allowed to define profiles to RACF for the classes that you specify. Classes you can specify are USER and any resource class name defined in the user profile. RACF deletes the class names you specify from the class names previously allowed for this user.

To enter the NOCLAUTH operand specifying a class in the class descriptor table, you must have the SPECIAL attribute, or the user's profile must be within the scope of a group in which you have the group-SPECIAL attribute and have the CLAUTH attribute, or you must be the owner of the user's profile and have the CLAUTH attribute for the class to be deleted.

To enter the NOCLAUTH operand specifying a class that is not in the class descriptor table you must have the SPECIAL attribute.

If you do not have sufficient authority for a specified class, RACF ignores the CLAUTH or NOCLAUTH specification for the class and continues processing with the next class name specified.

DATA | NODATA

DATA('installation-defined-data')

Specifies up to 255 characters of installation-defined data to be stored in the user's profile and must be enclosed in quotes. It might also contain double-byte character set (DBCS) data. Note that only 254 characters of data are available for installation exits. If your installation has exits that examine this data, you should specify a maximum of 254 characters.

Use the LISTUSER command to list this information.

NODATA

specifies that the ALTUSER command is to delete the installation-defined data in the user's profile.

DCE

Specifies that, when you define an z/OS DCE user to RACF, you can enter any of the following suboperands to specify information for that user. Each suboperand defines information that RACF stores in a field within the DCE segment of the user's profile.

You can control access to an entire DCE segment or to individual fields within the DCE segment by using field level access checking.

AUTOLOGIN(YES | NO) | **NOAUTOLOGIN**

Specifies whether z/OS UNIX DCE is to log this user into z/OS UNIX DCE automatically. If AUTOLOGIN(NO) or NOAUTOLOGIN is specified, z/OS UNIX DCE does *not* attempt to login this user to z/OS UNIX DCE automatically. If AUTOLOGIN is not specified, AUTOLOGIN(NO) is the default.

DCENAME(*user-principal-name*) | **NODCENAME**

Specifies the DCE principal name defined for this RACF user in the DCE registry.

The DCENAME you define to RACF can contain 1-1023 characters and can consist of any character. You can enter the name with or without single quotes, depending on the following:

ALTUSER

- If parentheses, commas, blanks, or semicolons are entered as part of the name, the character string must be enclosed in single quotes.
- If a single quote is intended to be part of the character string, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered. RACF does not ensure that a valid DCENAME has been specified.

The DCENAME assigned to a user must be the same as the DCE principal name defined to the DCE registry.

If DCENAME is not specified, the LISTUSER command does not display a DCENAME for this user.

Note: RACF does not enforce the uniqueness of each DCENAME. The DCENAME specified must match the user's DCE principal name that is defined to the DCE registry. If the DCENAME entered does not correspond to the DCE principal name entered in the DCE registry for this user, z/OS UNIX DCE cannot correctly associate the identity of the DCE principal with the correct RACF user ID.

NODCENAME

Specifies that you want to delete the DCE principal name from the DCE segment of the user's profile.

If NODCENAME is specified, the LISTUSER command does not display a DCENAME for this user.

HOMECELL(*dce-cell-name*) | NOHOMECELL

Specifies the DCE cell name defined for this RACF user.

The HOMECELL you define to RACF can contain 1-1023 characters and can consist of any character. You can enter the name with or without single quotes, depending on the following:

- If parentheses, commas, blanks, or semicolons are entered as part of the cell name, the character string must be enclosed in single quotes.
- If a single quote is intended to be part of the cell name, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered. The fully qualified path name should be specified. RACF does not ensure that a valid DCE cell name has been specified.

The HOMECELL assigned to a user **must** be the same as the DCE cell name that this user has been defined to.

If the HOMECELL is not specified, z/OS UNIX DCE single signon to DCE support assumes that the HOMECELL for this user is the same cell that this MVS system is defined to.

RACF checks that the prefix of the HOMECELL name entered has a prefix of either */.../* or */./*.

The notation */.../* indicates that the HOMECCELL name is a global domain name service (DNS) cell name or X.500 global name.

The notation */./* indicates that the HOMECCELL name is a cell relative CDS (cell directory service) name. When determining the naming conventions used within your DCE cell, you should contact your DCE cell administrator.

NOHOMECCELL

Specifies that you want to delete the cell information from the DCE segment of the user profile.

If NOHOMECCELL is specified, the LISTUSER command does not display the HOMECCELL for this user.

HOMEUUID(*home-cell-universal-unique-identifier*) | NOHOMEUUID

Specifies the DCE universal unique identifier (UUID) for the cell that this user is defined to. The UUID is a 36-character string that consists of numeric and hexadecimal characters. This string must have the delimiter of "-" in character positions 9, 14, 19, and 24. The general format for the UUID string is xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx, in which x represents a valid numeric or hexadecimal character.

Be careful when assigning UUIDs. The UUID *cannot* be randomly assigned. The HOMEUUID is the DCE UUID of the cell that this RACF user is defined to. If HOMEUUID is not specified, the LISTUSER command displays NONE for the HOMEUUID field.

Note: The HOMEUUID specified must match the UUID of the DCE cell to which this principal (specified by the DCENAME operand) is defined.

NOHOMEUUID

Specifies that you want to delete the home cell unique universal identifier from the DCE segment of the user's profile.

If NOHOMEUUID is specified, LISTUSER for that user ID shows NONE for the HOMEUUID field.

UUID(*universal-unique-identifier*) | NOUUID

Specifies the DCE universal unique identifier (UUID) of the DCE principal defined in DCENAME. The UUID is a 36-character string that consists of numeric and hexadecimal characters. This string must have the delimiter of "-" in character positions 9, 14, 19, and 24. The general format for the UUID string is xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx, in which x represents a valid numeric or hexadecimal character.

Be careful when assigning UUIDs. The UUID *cannot* be randomly assigned.

The DCE UUID assigned to a user **must** be the same as the DCE UUID assigned when defining this RACF user to the DCE registry as a DCE principal.

If UUID is not specified, the user cannot become an z/OS DCE user and a LISTUSER command for that user ID shows NONE for the UUID.

Note: RACF does not enforce the uniqueness of each UUID entered. The UUID specified must match the UUID in the DCE registry for the principal (specified by the DCENAME operand) that is being cross-linked with this RACF user ID.

ALTUSER

NOUUID

Specifies that you want to delete the DCE unique universal identifier from the DCE segment of the user's profile.

If NOUUID is specified, LISTUSER for that user ID shows NONE for the UUID field.

NODCE

Specifies that RACF should delete the DCE segment from the user's profile.

DFLTGRP(*group-name*)

Specifies the name of a RACF-defined group to be used as the new default group for the user. The user must already be connected to this new group with at least USE authority. The user remains connected to the previous default group.

DFP | NODFP

DFP

Specifies that when you change the profile of a user, you can enter any of the following suboperands to add, change, or delete default values for the DFP data application, data class, management class, and storage class. DFP uses this information to determine data management and DASD storage characteristics when a user creates a new data set.

You can control access to the entire DFP segment or to individual fields within the DFP segment by using field-level access checking. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

DATAAPPL | NODATAAPPL

DATAAPPL(*application-name*)

Specifies the name of a DFP data application. The name you specify can contain up to 8 alphanumeric characters.

NODATAAPPL

Specifies that you want to delete the DFP data application name from the DFP segment of the user's profile.

DATACLAS | NODATACLAS

DATACLAS(*data-class-name*)

Specifies the default data class. The class name you specify can contain up to 8 alphanumeric characters.

A data class can specify some or all of the physical data set attributes associated with a new data set. During new data set allocation, data management uses the value you specify as a default unless it is preempted by a higher priority default, or overridden in some other way (for example, by JCL).

The value you specify must be a valid data class name defined for use on your system. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

For information on defining DFP data classes, see *z/OS DFSMSdfp Storage Administration Reference*.

NODATACLAS

Specifies that you want to delete the default data class name from the DFP segment of the user's profile.

MGMTCLAS | NOMGMTCLAS

MGMTCLAS(*management-class-name*)

Specifies the default management class. The class name you specify can contain up to 8 alphanumeric characters.

A management class contains a collection of management policies that apply to data sets. Data management uses the value you specify as a default unless it is preempted by a higher priority default, or overridden in some other way (for example, by JCL).

The value you specify must be defined as a profile in the MGMTCLAS general resource class, and the user must be granted at least READ access to the profile. Otherwise, RACF does not allow the user access to the specified MGMTCLAS. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

For information on defining DFP management classes, see *z/OS DFSMSdfp Storage Administration Reference*.

NOMGMTCLAS

Specifies that you want to delete the default management class name from the DFP segment of the user's profile.

STORCLAS | NOSTORCLAS**STORCLAS**(*storage-class-name*)

Specifies the default storage class. The class name you specify can contain up to 8 alphanumeric characters.

A storage class specifies the service level (performance and availability) for data sets managed by the storage management subsystem (SMS). During new data set allocation, data management uses the value you specify as a default unless it is preempted by a higher priority default, or overridden in some other way (for example, by JCL).

The value you specify must be defined as a profile in the STORCLAS general resource class, and the user must be granted at least READ access to the profile. Otherwise, RACF does not allow the user access to the specified STORCLAS. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

For information on defining DFP storage classes, see *z/OS DFSMSdfp Storage Administration Reference*.

NOSTORCLAS

Specifies that you want to delete the default storage class name from the DFP segment of the user's profile.

NODFP

Specifies that RACF should delete the DFP segment from the user's profile.

EIM | NOEIM

Specifies or deletes the bind information required to establish a connection with the EIM domain.

EIM

Specifies the EIM segment for the user's profile.

LDAPPROF(*ldapbind_profile*)

Specifies the name of a profile in the LDAPBIND class. The profile in the LDAPBIND class contains the name of an EIM domain and the bind

ALTUSER

information required to establish a connection with the EIM domain. The EIM services will attempt to retrieve this information when it is not explicitly supplied via invocation parameters. Applications or other services which use the EIM services, may instruct their invokers to define a profile in the LDAPBIND class or the IRR.PROXY.DEFAULTS profile in the FACILITY class.

The *ldapbind_profile* specifies the name of a profile in the LDAPBIND class containing the EIM domain and the LDAP bind information. The *ldapbind_profile* name may be 1 to 246 characters long. It is not a case-sensitive name.

NOLDAPPROF

Deletes the LDAPBIND profile name from the EIM segment in the user's profile.

NOEIM

Deletes the EIM segment from the user's profile

EXPIRED | NOEXPIRED

EXPIRED

Specifies that the new password specified or defaulted by the PASSWORD keyword be marked as expired. This requires the user to change the password at the next logon or job start.

The EXPIRED keyword is only valid when specified with the PASSWORD keyword.

NOEXPIRED

Specifies that the password specified by the PASSWORD keyword does not need to be changed at the next logon. The NOEXPIRED keyword is only valid when specified with the PASSWORD keyword. NOEXPIRED does **not** indicate that the password never expires. If you wish to set a password that never expires, use the NOINTERVAL keyword on the PASSWORD command.

When NOEXPIRED is specified, the value supplied or defaulted by the PASSWORD operand is subject to rules set by both the installation (through the SETROPTS PASSWORD command) and the new-passowrd exit, ICHPWX01.

To specify NOEXPIRED you must either have the SPECIAL attribute (at the system level), or have UPDATE access to the IRR.PASSWORD.RESET resource in the FACILITY class. Being the owner of the USER profile or having the group-SPECIAL attribute is **not** sufficient when NOEXPIRED is specified.

GROUP(group-name)

Specifies the group to which changes to the group-related user attributes UACC and AUTHORITY are to be made. The user must be connected to the specified group.

If you omit GROUP, the changes apply to the user's default group. If you omit GROUP and specify DFLTGRP, however, the changes still apply to the user's previous default group.

GRPACC | NOGRPACC

GRPACC

Specifies that any group data sets protected by DATASET profiles defined by this user are automatically accessible to other users in the group. The

group whose name is used as the high-level qualifier of the data set name (or the qualifier supplied by a command installation exit) has UPDATE access authority in the new profile. GRPACC specified on the ALTUSER command overrides NOGRPACC specified on the CONNECT command.

NOGRPACC

Specifies that the user no longer has the GRPACC attribute.

KERB | NOKERB

KERB

Specifies Security Server Network Authentication Service information for a user defined to RACF. Each subkeyword defines information that RACF stores in a field within the KERB segment of the user's profile.

Note: The RACF user password must be changed to be non-expired in order to complete the definition of the Network Authentication Service principal. The user cannot use any Network Authentication Service function until the definition is complete.

ENCRYPT [[(DES | NODES) [DES3 | NODES3] [DESD | NODESD]]] NOENCRYPT

The ENCRYPT values are used to specify which keys are allowed for use based on the encryption algorithm used to generate them. The default values for ENCRYPT are DES, DES3, and DESD. You can use the following values to specify which keys are allowed for use by a principal.

DES	DES encrypted keys are allowed for use.
NODES	No DES encrypted keys are allowed for use.
DES3	DES3 encrypted keys are allowed for use.
NODES3	No DES3 encrypted keys are allowed for use.
DESD	DESD encrypted keys are allowed for use.
NODESD	No DESD encrypted keys are allowed for use.

The values in effect are dependent on the current SETROPTS KERBLVL setting.

When SETROPTS KERBLVL(0) is in effect the ENCRYPT settings will be ignored. Regardless of the settings DES keys will be generated and processed.

When SETROPTS KERBLVL(1) is in effect, or when SETROPTS KERBLVL gets changed from 0 to 1, the ENCRYPT settings will go into effect. Therefore, on password change, all three keys are generated and stored in the user's profile. The ENCRYPT setting will be used to determine which keys can be processed.

If you do not want to accept the defaults, you must specify the values you desire. For example, if you want to use only DES3 encryption, you must specify ENCRYPT(NODES DES3 NODESD).

If you specify ENCRYPT(NODES, NODES3, NODESD) at KERBLVL(1), no keys can be used, but all three will be generated and stored. At KERBLVL(0), the DES key will still be generated and it cannot be disallowed.

NOENCRYPT

Specifies that there is no restriction on which generated keys are to be

ALTUSER

allowed, and resets KERB encryption to the default settings. The NOENCRYPT operand has no effect at KERBLVL(0).

KERBNAME(*kerberos-principal-name*) | NOKERBNAME

KERBNAME(*kerberos-principal-name*)

Specifies the z/OS user ID's local *kerberos-principal-name*.

The value specified for the local *kerberos-principal-name* must be unique. Consequently, a list of users cannot be specified on an ALTUSER command with the KERBNAME keyword.

The *kerberos-principal-name* you define to RACF can consist of any character except the @ (X'7C') character. It is highly recommended that you avoid using **any** of the EBCDIC variant characters be avoided to prevent problems between different code pages. You can enter the name with or without single quotes, depending on the following:

- If parentheses, commas, blanks, or semicolons are entered as part of the name, the name must be enclosed in single quotes.
- If a single quote is intended to be part of the name and the entire character string is enclosed in single quotes, you must use two single quotes together to represent each single quote within the string.
- If the first character of the name is a single quote, you must enter the string within single quotes, with two single quotes entered for that single quote.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. However, RACF does not ensure that a valid *kerberos-principal-name* has been specified.

A local *kerberos-principal-name* **must not** be qualified with a realm name when specified with the KERBNAME keyword. However, RACF verifies that the local principal name, when fully qualified with the name of the local realm:

`/.../local_realm_name/principal_name`

does not exceed 240 characters. For example,

- If the local realm name is

`X`

fully qualified local principal names are prefixed with

`/.../X/`

and are limited to a maximum of 233 characters.

- If the local realm name is

`KERB390.ENDICOTT.IBM.COM`

fully qualified local principal names are prefixed with

`/.../KERB390.ENDICOTT.IBM.COM/`

and are limited to a maximum of 210 characters.

This length verification requires that the REALM profile for the local realm KERBDFLT be defined and contain the name of the local realm, prior to the specification of local Network Authentication Service user principals. Otherwise, Network Authentication Service users will not be defined.

Note: Because of the relationship between realm names and local *kerberos-principal-names*, in which the length of a fully qualified name cannot exceed 240 characters, caution and planning must go into renaming the local realm since the combined length is only checked by RACF when a local *kerberos-principal-name* is added or altered. Renaming the realm should be avoided as a result.

NOKERBNAME

deletes the *kerberos-principal-name*. This invalidates the z/OS user ID's Network Authentication Service account.

MAXTKTLFE(max-ticket-life) | NOMAXTKTLFE

MAXTKTLFE(max-ticket-life)

Specifies the *max-ticket-life* in seconds, and is represented by a numeric value between 1 and 2 147 483 647. Note that 0 is not a valid value.

If MAXTKTLFE is specified on the definition of a local Network Authentication Service principal, the Security Server Network Authentication Service takes the most restrictive of the value defined for the local principal and the value specified on the definition of the local realm (the KERBDFLT profile in the REALM class). Consequently, if the realm *max-ticket-life* is 24 hours, a principal cannot get a ticket with a longer lifetime even if the *max-ticket-life* is set to 48 hours. If this field is not specified for a local principal, or if NOMAXTKTLFE has been specified, the maximum lifetime for tickets created by this principal is determined from the definition of the local Network Authentication Service realm.

NOMAXTKTLFE

Deletes the *max-ticket-life* value for this local Network Authentication Service principal.

NOKERB

Deletes the user's KERB segment. This user is no longer considered a principal by the Security Server Network Authentication Service.

LANGUAGE| NOLANGUAGE

Specifies to add, alter, or delete the user's preferred national languages.

Specify LANGUAGE if this user is to have languages other than the ones established or defaulted by the LANGUAGE operand on the SETROPTS command, or the ones previously specified with the ADDUSER command.

LANGUAGE(PRIMARY(language) SECONDARY(language))

Specifies the user's preferred national languages. Specify this operand if the user is to have languages other than the system-wide defaults (established by the LANGUAGE operand on the SETROPTS command).

- If this profile is for a TSO/E user who will establish an extended MCS console session, the languages you specify should be one of the

ALTUSER

languages specified on the LANGUAGE LANGCODE statements in the MMSLSTxx PARMLIB member. See your MVS system programmer for this information.

For more information on TSO/E national language support, see *z/OS TSO/E Customization*.

- If this profile is for a CICS user, see your CICS administrator for the languages supported by CICS on your system.

For more information, see *CICS RACF Security Guide*.

PRIMARY | NOPRIMARY

PRIMARY(*language*)

Specifies the user's new primary language.

NOPRIMARY

Deletes any primary language information from the user's profile and returns the user to the installation's default primary language.

SECONDARY | NOSECONDARY

SECONDARY(*language*)

Specifies the language to which the user's secondary language is to be changed.

NOSECONDARY

Deletes any secondary language information from the user's profile and returns the user to the installation's default secondary language.

Notes:

1. For the primary and secondary languages, specify either the installation-defined name of a currently active language (a maximum of 24 characters) or one of the language codes (three characters in length) for a language installed on your system.
2. The language name can be a quoted or unquoted string.
3. The same language can be specified for with both PRIMARY and SECONDARY parameters.
4. If the MVS message service is not active, the PRIMARY and SECONDARY values must be a 3 character language code.

NOLANGUAGE

Deletes the user's preferred national languages from the profile and returns that user to the installation defaults. LANGUAGE information no longer appears in LISTUSER output.

LNOTES | NOLNOTES

LNOTES

Specifies Lotus Notes for z/OS information for the user profile being changed.

SNAME | NOSNAME

SNAME(*short-name*)

Specifies the Lotus Notes for z/OS *short-name* of the user being changed. The name should match the one stored in the Lotus Notes address book for this user, but this is not verified by the command.

The *short-name* you define to RACF can contain 1-64 characters. You can specify the following characters: upper and lower case alphabets (A through Z, and a through z), 0 through 9, & (X'50'), - (X'60'), . (X'4B'), _ (X'6D'), and (X'40').

If the *short-name* you specify contains any blanks, it must be enclosed in single quotes. The *short-name* is stripped of leading and trailing blanks.

The value specified for the *short-name* must be unique. Consequently, a list of users might not be specified on an ALTUSER command with the SNAME keyword.

NOSNAME

Specifies that you want to delete the *short-name* from the LNOTES segment of the user's profile.

NOLNOTES

Specifies that you want to delete the LNOTES segment from the user's profile.

MODEL | NOMODEL

MODEL(*dsname*)

Specifies the name of a data set that RACF is to use as a model when new data set profiles are created that have *userid* as the high-level qualifier. For this operand to be effective, the MODEL(USER) option (specified on the SETROPTS command) must be active. If the ALTUSER command cannot find the *dsname* profile, it issues a warning message but places the model name in the userid entry.

Note that RACF always prefixes *dsname* with the user ID.

For information about automatic profile modeling, refer to *z/OS Security Server RACF Security Administrator's Guide*.

NOMODEL

Deletes the model profile name in the user's profile.

NAME(*user-name*)

Specifies the user name to be associated with the user ID. You can use a maximum of 20 alphanumeric or non-alphanumeric characters. If the name you specify contains any blanks, it must be enclosed in single quotes.

If you omit the NAME operand, RACF uses a default of 20 # (X'7B') characters ('###...'). Note, however, that the corresponding entry in a LISTUSER output is the word "unknown".

NDS | NONDS

NDS

Specifies Novell Directory Services for OS/390 information for the user profile being changed.

UNAME | NOUNAME

UNAME(*user-name*)

Specifies the Novell Directory Services for OS/390 *user-name* of the user being changed. The *user-name* value should match the name stored in the Novell Directory Services for OS/390 directory for this user, but this is not verified by the command.

The *user-name* you define to RACF can contain 1-246 characters. However, the *user-name* can not contain the following characters: *

ALTUSER

(X'5C'), + (X'4E'), | (X'4F'), = (X'7E'), , (X'6B'), " (X'7F'), ` (X'79'), / (X'61'), : (X'7A'), ; (X'5E'), ¢ (X'4A'), and brackets ([and], X'AD' and X'BD' respectively).

If the *user-name* you specify contains any parentheses or blanks, it must be enclosed in single quotes. The *user-name* is stripped of leading and trailing blanks. If a single quotation mark is intended to be part of the *user-name*, you must use two single quotation marks together for each single quotation mark within the string, and the entire string must then be enclosed within single quotation marks.

The value specified for the *user-name* must be unique. Consequently, a list of users cannot be specified on an ALTUSER command with the UNAME keyword.

NOUNAME

Specifies that you want to delete the *user-name* from the NDS segment of the user's profile.

NONDS

Specifies that RACF delete the NDS segment from the user's profile.

NETVIEW | NONETVIEW

NETVIEW

Specifies that this is a NetView operator that can enter any of the following suboperands to add, update, or delete the information in the NETVIEW segment.

You can control access to the entire NETVIEW segment or to individual fields within the NETVIEW segment by using field-level access checking. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

CONSNAME | NOCONSNAME

CONSNAME(console-name)

Specifies the default MCS console name identifier used for this operator. This default console name is used when the operator does not specify a console name using the NetView GETCONID command.

console-name is a 1-8 character identifier whose validity is checked by MVS processing when the operator tries to use it. See *z/OS MVS Planning: Operations* for information on valid values for a particular release.

NOCONSNAME

Deletes any default MCS console name previously specified for this operator.

CTL | NOCTL

CTL (GENERAL | GLOBAL | SPECIFIC)

Specifies whether a security check is performed for this NetView operator when they try to use a span or try to do a cross-domain logon.

GENERAL

Specifies that a security done should be done as for SPECIFIC, and, in addition, that the operator is allowed to access devices that are not part of any span.

GLOBAL

Specifies that no security check is done.

SPECIFIC

Specifies that a security check is performed through RACROUTE REQUEST=AUTH whenever this operator attempts to use a span. It also specifies that any cross-domain logon must be to a domain listed in the operator's NETVIEW segment with the DOMAINS keyword.

CTL(SPECIFIC) is the default.

NOCTL

NOCTL has the same effect as specifying CTL(SPECIFIC).

DOMAINS | NODOMAINS | ADDDOMAINS | DELDOMAINS**DOMAINS**(*domain-name* ...)

Specifies the complete list of identifiers of NetView programs in another NetView domain where this operator can start a cross-domain session. The NetView program identifiers are coded on the NCCFID definition statement for the other domains, and represent the name given to that NetView program on the APPL statement.

Domain-name is a 1-5 character identifier. The characters can be alphabetic, numeric, or national.

ADDDOMAINS(*domain-name* ...)

Adds identifiers of NetView programs in another NetView domain where this operator can start a cross-domain session. The NetView program identifiers are coded on the NCCFID definition statement for the other domains, and represent the name given to that NetView program on the APPL statement.

domain-name is a 1-5 character identifier. The characters can be alphabetic, numeric, or national.

DELDOMAINS(*domain-name* ...)

Deletes specific identifiers of NetView programs in another NetView domain where this operator can start a cross-domain session. The NetView program identifiers are coded on the NCCFID definition statement for the other domains, and represent the name given to that NetView program on the APPL statement.

domain-name is a 1-5 character identifier. The characters can be alphabetic, numeric, or national.

NODOMAINS

Specifies that the operator cannot start any cross-domain sessions.

IC | NOIC**IC**('command | command-list')

specifies the command or command list (up to 255 characters) to be processed when the operator logs on to NetView.

If the command or command list you specify contains any commas, blanks, or other special characters that TSO/E requires to be quoted, it must be enclosed in single quotes.

NOIC

Deletes the command or command list to be processed at logon time for this operator. No command or command list is automatically processed when this operator logs on.

MSGRECVR | NOMSGRECVR

ALTUSER

MSGRECVR (NO | YES)

Specifies whether this operator can receive unsolicited messages that are not routed to a specific NetView operator.

NO

Specifies that the operator is not to receive the messages.

YES

Specifies that the operator is to receive the messages.

NOMSGRECVR

NOMSGRECVR has the same effect as specifying MSGRECVR(NO).

NGMFADMN | NONGMFADMN

NGMFADMN (NO | YES)

Specifies whether a NetView operator has administrator authority to the NetView Graphic Monitor Facility (NGMF).

NO

Specifies that the operator does not have the authority.

YES

Specifies that the operator does have the authority.

NONGMFADMN

NONGMFADMN has the same effect as specifying NGMFADMN(NO).

NGMFVSPN | NONGMFVSPN

NGMFVSPN (*view-span*)

Reserved for future use by the NetView Graphic Monitor Facility

NONGMFVSPN

Reserved for future use by the NetView Graphic Monitor Facility

OPCLASS | NOOPCLASS | ADDOPCLASS | DELOPCLASS

OPCLASS(*class ...*)

Specifies the complete list of NetView scope classes for which the operator has authority.

class is a number from 1 to 2040 that specifies a NetView scope class.

ADDOPCLASS(*class ...*)

Adds specific NetView scope classes to the operator's current list of classes.

class is a number from 1 to 2040 that specifies a NetView scope class.

DELOPCLASS(*class ...*)

Deletes specific NetView scope classes from the operator's current list of classes.

class is a number from 1 to 2040 that specifies a NetView scope class.

NOOPCLASS

Specifies that the operator is in no scope classes.

NONETVIEW

Specifies that RACF should delete the NETVIEW segment from the user's profile.

OIDCARD | NOOIDCARD

OIDCARD

Specifies that the user must supply an operator identification card when

logging onto the system. If you specify the OIDCARD operand, the system prompts you to enter the user's new operator identification card as part of the processing of the ALTUSER command. If you specify the OIDCARD operand in a job executing in the background or when you cannot be prompted in the foreground, the ALTUSER command fails.

NOOIDCARD

Specifies that the user is not required to supply an operator identification card.

If NOPASSWORD is specified or the user ID already has the NOPASSWORD attribute, specifying NOOIDCARD causes this user ID to become a protected user ID. Protected user IDs cannot be used to enter the system by any means that requires a password to be specified, such as TSO logon. If the user attempts to enter the system with a password, the attempt fails.

Protected user IDs can be used for the user IDs associated with the started tasks in ICHRIN03 or the STARTED class.

OMVS | NOOMVS

OMVS

Specifies z/OS UNIX information for the user profile being changed.

You can control access to the entire OMVS segment or to individual fields within the OMVS segment by using field-level access checking.

ASSIZEMAX | NOASSIZEMAX

ASSIZEMAX(*address-space-size*)

Specifies the RLIMIT_AS hard limit (maximum) resource value that processes receive when they are dubbed a process. The *address-space-size* you define to RACF is a numeric value between 10 485 760 and 2 147 483 647. ASSIZEMAX indicates the address space region size in bytes. The soft limit (current) resource value is obtained from MVS. If the soft limit value from MVS is greater than the address space size, the soft limit is used.

The value specified for ASSIZEMAX is also used when processes are initiated by a daemon process using an exec after `setuid()`. In this case, both the RLIMIT_AS hard limit and soft limit are set to the *address-space-size* value.

The value specified for ASSIZEMAX overrides any value provided by the MAXASSIZE parameter of BPXPRMxx. For more information, see *z/OS UNIX System Services Planning*.

NOASSIZEMAX

Specifies that you want to delete the address space size from the OMVS segment of the user's profile. The value specified for MAXASSIZE in BPXPRMxx now applies to the user.

AUTOUID | UID | NOUID

Specifies whether RACF is to automatically assign an unused UID value to the user, assign a specific UID value or delete the user identifier from the OMVS segment of the user's profile.

AUTOUID

Specifies that RACF is to automatically assign an unused UID value to the user. The UID value is derived from information obtained from the BPX.NEXT.USER profile in the FACILITY class. For more

ALTUSER

information on setting up BPX.NEXT.USER, see *z/OS Security Server RACF Security Administrator's Guide*.

If you are using RRSF automatic command direction for the USER class, the command sent to other nodes will contain an explicit assignment of the UID value which was derived by RACF on the local node.

Rules:

- AUTOUID cannot be specified if more than one user ID is entered.
- The AUTOUID keyword is mutually exclusive with the SHARED keyword.
- If both UID and AUTOUID are specified, AUTOUID is ignored.
- If both NOUID and AUTOUID are specified, AUTOUID is ignored.
- Field- level access checking for the UID field applies when using AUTOUID.
- AUTOUID cannot be used to reassign a UID value when one already exists for the user. If AUTOUID is specified, but the user already has a UID assigned, one of two things will happen.
 - If the pre-existing UID is unique to this user, this value will be identified in informational message IRR52177I, and the value will be left unchanged. If RACF Remote Sharing Facility (RRSF) automatic command direction is in effect for the USER class, then the outbound ALTUSER command will be altered to contain the pre-existing UID value in the OMVS UID keyword.
 - If the pre-existing UID is not unique to this user, error message IRR52178I will be issued, and the command will fail. See IRR52178I for information on changing the user's existing UID value.

UID(*user-identifier*)[SHARED]

UID(*user-identifier*)

Specifies the user identifier. The UID is a numeric value between 0 and 2 147 483 647.

When assigning a UID to a user, you should make sure that the user's default group has a GID. A user who has a UID and a current connect group that has a GID can use functions such as the TSO/E OMVS command and can access HFS files based on the UID and GID values assigned.

Care should be taken in assigning 0 as the user identifier. UID 0 is considered a superuser. The superuser passes all z/OS UNIX security checks. Assigning a UID to a user ID that appears in the RACF started procedures table (ICHRIN03) should also be done with care. RACF defined started tasks that have the trusted or privileged attribute are considered superusers even if their UID is a value other than 0.

If the UID is not specified, the user is unable to become a z/OS UNIX user and a LISTUSER for that user ID shows NONE for the UID.

Notes:

1. If the security administrator has defined the SHARED.IDS profile in the UNIXPRIV class, the UID value must be unique. Use the SHARED keyword in addition to UID to assign a value that is already in use.
2. If SHARED.IDS is not defined, RACF does not require the UID to be unique. The same value can be assigned to multiple users but this is not recommended because individual user control would be lost. However, if you want a set of users to have exactly the same access to z/OS UNIX resources, you might decide to assign the same UID to more than one user.

SHARED

If the security administrator has chosen to control the use of shared UIDs, this keyword must be used in addition to the UID keyword to specify the user identifier if it is already in use by at least one other user. The administrator controls shared UIDs by defining the SHARED.IDS profile in the UNIXPRIV class.

Rules:

- If the SHARED.IDS profile is not defined, SHARED is ignored.
- If SHARED is specified in the absence of UID, it is ignored.
- If the SHARED.IDS profile is defined and SHARED is specified, but the value specified with UID is not currently in use, SHARED is ignored and UNIXPRIV authority is not required.
- Field-level access checking for the UID field applies when using SHARED.
- The SHARED keyword is mutually exclusive with the AUTOUID keyword.

NOUID

Specifies that you want to delete the user identifier from the OMVS segment of the user's profile.

If NOUID is specified, the user is unable to become a z/OS UNIX System Services user and a LISTUSER for that user ID shows NONE for the UID.

CPUTIMEMAX | NOCPUTIMEMAX**CPUTIMEMAX(*cpu-time*)**

Specifies the RLIMIT_CPU hard limit (maximum) resource value that the user's z/OS UNIX processes receive when they are dubbed a process. The *cpu-time* you define to RACF is a numeric value between 7 and 2 147 483 647. RLIMIT_CPU indicates the *cpu-time* that a process is allowed to use, in seconds. The soft limit (current) is obtained from MVS. If the soft limit (current) resource value from MVS is greater than the *cpu-time* value, the soft limit is used.

ALTUSER

The value specified for CPUTIMEMAX is also used when processes are initiated by a daemon process using an `exec` after `setuid()`. In this case, both the RLIMIT_CPU hard and soft limits are set to the *cpu-time* value.

For processes running in, or forked from TSO or BATCH, the *cpu-time* value has no effect. For processes created by the `rlogin` command or other daemons, *cpu-time* is the time limit for the address space.

The value specified for CPUTIMEMAX overrides any value provided by the MAXCPUPTIME parameter of BPXPRMxx. For more information, see *z/OS UNIX System Services Planning*.

NOCPUTIMEMAX

Specifies that you want to delete the *cpu* time from the OMVS segment of the user's profile. The value specified for MAXCPUPTIME in BPXPRMxx now applies to the user.

FILEPROCMAX | NOFILEPROCMAX

FILEPROCMAX(*files-per-process*)

Specifies the maximum number of files the user is allowed to have concurrently active or open. The *files-per-process* you define to RACF is a numeric value between 3 and 262 143. FILEPROCMAX is the same as the OPEN_MAX variable defined in the POSIX standard.

FILEPROCMAX lets you limit the amount of system resources available to a user process. Select FILEPROCMAX by considering:

- For conformance to standards, set FILEPROCMAX to:
 - At least 16 to conform to the POSIX standard
 - At least 25 to conform to the FIPS standard
- 256 is a commonly recommended value.
- A process can change its own value for the number of files it has active or open using the `setrlimit()` function. Only processes with appropriate privileges can increase their limits.
- The minimum value of 3 supports the standard files for a process: `stdin`, `stdout`, and `stderr`.
- The value needs to be larger than 3 to support z/OS UNIX shell users. If the value is too small, the z/OS UNIX shells might issue the message "File descriptor not available."

The value specified for FILEPROCMAX overrides any value provided by the MAXFILEPROC parameter of BPXPRMxx. For more information, see *z/OS UNIX System Services Planning*.

NOFILEPROCMAX

Specifies that you want to delete the files per process from the OMVS segment of the user's profile. The value specified for MAXFILEPROC in BPXPRMxx now applies to the user.

HOME | NOHOME

HOME(*directory-pathname*)

Specifies the hierarchical file system (HFS) directory pathname. The directory is part of the file system. This is the current working directory for the user's process when the user enters the TSO/E command OMVS.

When you define a directory pathname to RACF, it can contain 1-1023 characters. The directory pathname can consist of any characters and can be entered with or without single quotes. The following rules apply:

- If parentheses, commas, blanks, or semicolons are to be entered as part of the pathname, the character string must be enclosed in single quotes.
- If a single quote is intended to be part of the pathname, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered. The fully-qualified pathname should be specified. RACF does not ensure that a valid pathname has been specified. If you issue the ALTUSER command as a RACF operator command and you specify the pathname in lowercase, you must include the pathname within single quotations.

NOHOME

Specifies that you want to delete the initial directory pathname from the OMVS segment of the user's profile.

If no value is specified for HOME in the OMVS segment, MVS sets the working directory for the user to "/" (the root directory).

MMAPAREAMAX | NOMMAPAREAMAX**MMAPAREAMAX**(*memory-map-size*)

Specifies the maximum amount of data space storage, in pages, that can be allocated by this user for memory mappings of HFS files. Storage is not allocated until memory mappings are active. The *memory-map-size* you define to RACF is a numeric value between 1 and 16 777 216.

Use of memory map services consumes a significant amount of system memory. For each page (4KB) that is memory mapped, 96 bytes of ESQA are consumed when a file is not shared with any other users. When a file is shared by multiple users, each user after the first causes 32 bytes of ESQA to be consumed for each shared page. The ESQA storage is consumed when the mmap() function is invoked by the application program.

The value specified for MMAPAREAMAX overrides any value provided by the MAXMMAPAREA parameter of BPXPRMxx. For more information, see *z/OS UNIX System Services Planning*.

NOMMAPAREAMAX

Specifies that you want to delete the memory map size from the OMVS segment of the user's profile. The value specified for MAXMMAPAREA in BPXPRMxx now applies to the user.

PROCUSERMAX | NOPROCUSERMAX**PROCUSERMAX**(*processes-per-UID*)

Specifies the maximum number of processes this user is allowed to have active at the same time, regardless of how the process became an z/OS UNIX process. The *processes-per-UID* you define

ALTUSER

to RACF is a numeric value between 3 and 32 767.
PROCUSERMAX is the same as the CHILD_MAX variable defined in the POSIX standard.

PROCUSERMAX allows you to limit user activity to optimize performance. Select PROCUSERMAX by considering:

- For conformance to standards, set PROCUSERMAX to:
 - At least 16 to conform to the POSIX standard
 - At least 25 to conform to the FIPS standard
- A user with a UID of 0 is not limited by the PROCUSERMAX value because a superuser might need to be capable of logging on and using z/OS UNIX services to solve a problem.
- A low PROCUSERMAX value limits the number of concurrent processes that the user can run. A low value also limits the user's consumption of processing time, virtual storage, and other system resources.
- Some daemons run without UID 0, and can create many address spaces. In these cases, it is necessary to set the limit high enough for the daemon associated with this user ID to run all of its processes.

Though not recommended, the same OMVS UID can be given to more than one user ID. If users share a UID, you need to define a greater number for PROCUSERMAX. An example is the user ID defined for the default OMVS segment, specified by the FACILITY class profile BPX.DEFAULT.USER.

The value specified for PROCUSERMAX overrides any value provided by the MAXPROCUSER parameter of BPXPRMxx. For more information, see *z/OS UNIX System Services Planning*.

NOPROCUSERMAX

Specifies that you want to delete the processes per UID from the OMVS segment of the user's profile. The value specified for MAXPROCUSER in BPXPRMxx now applies to the user.

THREADSMAX | NOTHEADSMAX

THREADSMAX(*threads-per-process*)

Specifies the maximum number of pthread_created threads, including those running, queued, and exited but not detached, that this user can have concurrently active. The *threads-per-process* you define to RACF is a numeric value between 0 and 100 000. Specifying a value of 0 prevents applications run by this user from using the pthread_create service.

The value specified for THREADSMAX overrides any value provided by the MAXTHREADS parameter of BPXPRMxx. For more information, see *z/OS UNIX System Services Planning*.

NOTHEADSMAX

Specifies that you want to delete the threads per process from the OMVS segment of the user's profile. The value specified for MAXTHREADS in BPXPRMxx now applies to the user.

PROGRAM | NOPROGRAM

PROGRAM(*program-name*)

Specifies the PROGRAM pathname (z/OS UNIX shell program). This is the first program started when the TSO/E command OMVS is entered or when a batch job is started using the BPXBATCH program.

When you define a PROGRAM pathname to RACF, it can contain 1-1023 characters. The PROGRAM pathname can consist of any characters and can be entered with or without single quotes. The following rules apply:

- If parentheses, commas, blanks, or semicolons are to be entered as part of the pathname, the character string must be enclosed in single quotes.
- If a single quote is intended to be part of the pathname, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered. The fully-qualified pathname should be specified. RACF does not ensure that a valid pathname has been specified. If you issue the ALTUSER command as a RACF operator command and you specify the pathname in lowercase, you must include the pathname within single quotations.

NOPROGRAM

Specifies that you want to delete the z/OS UNIX System Services program pathname from the OMVS segment of the user's profile.

If no value is specified for PROGRAM in the OMVS segment, MVS gives control to the default z/OS UNIX shell program when the user issues the TSO/E command OMVS or starts a batch job using the BPXBATCH program.

For more information about the default z/OS UNIX shell program supplied with z/OS UNIX System Services, see *z/OS UNIX System Services Planning* and *z/OS UNIX System Services User's Guide*.

NOOMVS

Specifies that RACF delete the OMVS segment from the user's profile.

OPERATIONS | NOOPERATIONS**OPERATIONS**

Specifies that the user is to have authorization to do maintenance operations on all RACF-protected DASD data sets, tape volumes, and DASD volumes except those where the access list specifically limits the OPERATIONS user to an access authority that is less than the operation requires.

You establish the lower access authority for the OPERATIONS user with the PERMIT command. OPERATIONS on the ALTUSER command overrides NOOPERATIONS on the CONNECT command.

You must have the SPECIAL attribute to use the OPERATIONS operand.

NOOPERATIONS

Specifies that the user is not to have the OPERATIONS attribute.

ALTUSER

You must have the SPECIAL attribute to use the NOOPERATIONS operand.

OPERPARM | NOOPERPARM

Specifies or deletes default information used when this user establishes an extended MCS console session.

You can control access to the entire OPERPARM segment or to individual fields within the OPERPARM segment by using field-level access checking. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

For information on planning how to use OPERPARM segments, see *z/OS MVS Planning: Operations*.

Notes:

1. You need not specify every suboperand in an OPERPARM segment. In general, if you omit a suboperand, the default is the same as the default in the CONSOLxx PARMLIB member, which can also be used to define consoles.
2. If you specify MSCOPE or ROUTCODE but do not specify a value for them, RACF uses MSCOPE(*ALL) and ROUTCODE(NONE) to update the corresponding fields in the user profile. These values appear in listings of the OPERPARM segment of the user profile.
3. If you omit the other suboperands, RACF does not update the corresponding fields in the user's profile, and no value appears in listings of the OPERPARM segment of the profile.

ALTGRP(*alternate-console-group*) | NOALTGRP

Specifies the console group used in recovery.

The variable *alternate-console-group* can contain 1-8 characters, with valid characters being 0 through 9, A through Z, # (X'7B'), \$ (X'5B'), or @ (X'7C').

NOALTGRP deletes alternate console group information from this profile.

AUTH(MASTER | ALL | INFO | any others) | NOAUTH

Specifies or deletes this console's authority to issue operator commands.

If you omit this operand, RACF does not alter this field in the user's profile. If this field has not been added to the user's profile, an extended MCS console uses AUTH(INFO) when a session is established.

The console can have the following authorities:

MASTER

Allows this console to act as a master console, which can issue all MVS operator commands. This authority can only be specified by itself.

ALL

Allows this console to issue system control commands, input/output commands, console control commands, and informational commands. This authority can only be specified by itself.

INFO

Allows this console to issue informational commands. This authority can only be specified by itself.

CONS

Allows this console to issue console control and informational commands.

IO Allows this console to issue input/output and informational commands.

SYS

Allows this console to issue system control commands and informational commands.

NOAUTH

Deletes the user's operator authorities from the profile. Console operator authority no longer appears in profile listings. However, AUTH(INFO) is used when an extended MCS console session is established.

AUTO | NOAUTO**AUTO(YES | NO)**

Specifies whether the extended console can receive messages which have been automated by the Message Processing Facility (MPF) in the sysplex.

NOAUTO deletes this field from the user's profile. AUTO information no longer appears in profile listings. However, AUTO(NO) is used when an extended MCS console session is established.

CMDSYS | NOCMDSYS**CMDSYS(system-name | *)**

Specifies the system to which commands from this console are to be sent.

If you omit this operand, RACF does not alter this field in the user's profile. If this field has not been added to the user's profile, an extended MCS console uses CMDSYS(*) when a session is established. The variable *system-name* must contain 1-8 characters (A through Z, 0 through 9, and @ (X'7C'), # (X'7B'), \$ (X'5B')). If (*) is specified, commands are processed on the local system where the console is attached.

NOCMDSYS

Deletes any system-names from this profile. No CMDSYS information appears in profile listings. However, CMDSYS(*) is used when an extended MCS console session is established.

DOM | NODOM**DOM(NORMAL | ALL | NONE)**

Specifies which delete operator message (DOM) requests this console can receive.

If you omit this operand, RACF does not alter this field in the user's profile. If this field has not been added to the user's profile, an extended MCS console uses DOM(NORMAL) when a session is established.

NORMAL

The system queues all appropriate DOM requests to this console.

ALL

All systems in the sysplex queue DOM requests to this console.

NONE

No DOM requests are queued to this console.

ALTUSER

NODOM

Deletes this field from the user's profile. DOM information no longer appears in profile listings. However, DOM(NORMAL) is used when an extended MCS console session is established.

KEY | NOKEY

KEY(*searching-key*)

Specifies a 1-8 character name that can be used to display information for all consoles with the specified key by using the MVS command DISPLAY CONSOLES,KEY. If specified, KEY can include A through Z, 0 through 9, #(X'7B'), \$(X'5B'), or @(X'7C').

If you omit this operand, RACF does not alter this field in the user's profile. If this field has not been added to the user's profile, an extended MCS console uses a KEY value of NONE when a session is established.

NOKEY

Deletes search key information from the user's profile. Search key information no longer appears in profile listings. However, a KEY value of NONE is used when an extended MCS console session is established.

LEVEL | NOLEVEL

LEVEL(*message-level*)

Specifies the messages that this console is to receive.

If you omit this operand, RACF does not alter this field in the user's profile. If this field has not been added to the user's profile, an extended MCS console uses LEVEL(ALL) when a session is established.

The *message-level* variable can be a list of R, I, CE, E, IN, NB or ALL. If you specify ALL, you cannot specify R, I, CE, E, or IN.

- NB** The console receives *no* broadcast messages.
- ALL** The console receives these messages: R, I, CE, E, IN.
- R** The console receives messages requiring an operator reply.
- I** The console receives immediate action messages.
- CE** The console receives critical eventual action messages.
- E** The console receives eventual action messages.
- IN** The console receives informational messages.

NOLEVEL

Deletes any defined message levels for this console from the profile. Message information no longer appears in profile listings. However, LEVEL(ALL) is used when an extended MCS console session is established.

LOGCMDRESP | NOLOGCMDRESP

LOGCMDRESP(SYSTEM | NO)

Specifies if command responses are to be logged.

If you omit this operand, RACF does not alter this field in the user's profile. If this field has not been added to the user's profile, an extended MCS console uses LOGCMDRESP(SYSTEM) when a session is established.

SYSTEM

Command responses are logged in the hardcopy log.

NO

Command responses are not logged.

NOLOGCMDRESP

Deletes the value for LOGCMDRESP from the profile. Command response logging information no longer appears in profile listings. However, "LOGCMDRESP(SYSTEM)" is used when an extended MCS console session is established.

MFORM | NOMFORM**MFORM**(*message-format*)

Specifies the format in which messages are displayed at the console.

If you omit this operand, RACF does not alter this field in the user's profile. If this field has not been added to the user's profile, an extended MCS console uses MFORM(M) when a session is established.

The *message-format* variable can be a combination of T, S, J, M, and X.

- J** Messages are displayed with a job ID or name.
- M** The message text is displayed.
- S** Messages are displayed with the name of the originating system.
- T** Messages are displayed with a time stamp.
- X** Messages that are flagged as exempt from job name and system name formatting are ignored.

NOMFORM

Deletes the values for MFORM from the profile and causes message text to be displayed (MFORM(M)) when an extended MCS console session is established.

MIGID | NOMIGID**MIGID**(YES | NO)

Specifies whether a 1-byte migration ID is to be assigned to this console or not. The migration ID allows command processors that use a 1-byte console ID to direct command responses to this console.

If you omit this operand, RACF does not alter this field in the user's profile. If this field has not been added to the user's profile, an extended MCS console uses MIGID(NO) when a session is established.

NOMIGID

Deletes this segment from the profile. Migration identification information no longer appears in profile listings. However, MIGID(NO) is assigned when an extended MCS console session is established.

MONITOR | NOMONITOR**MONITOR(*events*)**

Specifies which information should be displayed when monitoring jobs, TSO sessions, or data set status.

If you omit this operand, RACF does not alter this field in the user's profile. If this field has not been added to the user's profile, an extended MCS console uses MONITOR(JOBNAMES SESS) when a session is established. The variable *events* can be a list of the following:

JOBNAMES | JOBNAMEST

Displays information about the start and end of each job.

JOBNAMES omits the times of job start and job end. JOBNAMEST displays the times of job start and job end.

SESS | SESST

Displays information about the start and end of each TSO session.

SESS omits the times of session start and session end. SESST displays the times of session start and session end.

STATUS

Specifies that the information displayed when a data set is freed or unallocated should include the data set status.

NOMONITOR

Deletes job monitor information from the user's profile. Information from this field no longer appears in profile listings. However, MONITOR(JOBNAMES SESS) is used when an extended MCS console session is established.

MSCOPE | ADDMSCOPE | DELMSCOPE | NOMSCOPE**MSCOPE(*system-name... | * | *ALL*)**

Specifies the systems from which this console can receive messages that are not directed to a specific console.

If you omit this operand, RACF does not alter this field in the user's profile. If this field has not been added to the user's profile, an extended MCS console uses MSCOPE(*ALL) when a session is established. If you specify MSCOPE but omit a value, RACF uses MSCOPE(*ALL) as the default to update this field in the user's profile. *ALL appears in listings of the OPERPARM segment of the user's profile.

system-name...

Is a list of one or more system names, where a system name can be any combination of A through Z, 0 through 9, # (X'7B'), \$ (X'5B'), or @ (X'7C').

* Is the system on which the console is currently active.

***ALL**

Means all systems.

ADDMSCOPE(*system-name...*)

Adds the specified system names to the existing list of systems from which this console can receive messages that are not directed to a specific console.

DELMSCOPE(system-name...)

Deletes the specified system names from the existing list of systems from which this console can receive messages that are not directed to a specific console.

NOMSCOPE

Deletes any system name information from the user's profile. Message reception information no longer appears in profile listings. However, MSCOPE(*ALL) is used when an extended MCS console session is established.

ROUTCODE(ALL | NONE | *routing-codes*) | NOROUTCODE**ROUTCODE(ALL | NONE | *routing-codes*)**

Specifies the routing codes of messages this operator is to receive.

If you omit this operand, RACF does not alter this field in the user's profile. If this field has not been added to the user's profile, an extended MCS console uses ROUTCODE(NONE) when a session is established. If you specify ROUTCODE but omit a value, RACF uses ROUTCODE(NONE) to update this field in the user's profile. NONE appears in listings of the OPERPARM segment of the user's profile.

The routing code information can be one of the following:

ALL

Means all routing codes.

NONE

Means no routing codes.

routing-codes

Specifies one or more routing codes or sequences of routing codes. The routing codes can be a list of *n* and *n1:n2*, where *n*, *n1*, and *n2* are integers from 1 to 128, and *n1:n2* represents a range of routing codes from *n1* (low) to *n2* (high).

NOROUTCODE

Deletes routing code information from the user's profile. Routing code information no longer appears in profile listings. However, ROUTCODE(NONE) is used when an extended MCS console session is established.

STORAGE | NOSTORAGE**STORAGE(*amount*)**

Specifies the amount of storage in the TSO/E user's address space that can be used for message queuing to this console.

If you omit this operand, RACF does not alter this field in the user's profile. If this field has not been added to the user's profile, an extended MCS console uses STORAGE(1) when a session is established. A value of 0 will appear in listings of the user's profile to indicate that no value was specified. The variable *amount* must be a value between 1 and 2000.

NOSTORAGE

Deletes this field from the profile. A value of 0 appears in listings of the

ALTUSER

user's profile to indicate that no value was specified. However, STORAGE(1) is used when an extended MCS console session is established.

UD | NOUD

UD(YES | NO)

Specifies whether this console is to receive undelivered messages.

If you do not specify this operand, RACF does not alter the user's profile. If this field has not been added to the user's profile, an extended MCS console uses UD(NO) when a session is established.

NOUD

Deletes the field from the profile. Undelivered message information no longer appears in profile listings. However, UD(NO) is used when an extended MCS console session is established.

NOOPERPARM

Specifies that the OPERPARM segment is to be deleted. Operator information no longer appears in LISTUSER output.

OVM | NOOVM

Specifies OpenExtensions VM information for the user profile being changed. Information is stored in the OVM segment of the user's profile.

You can control access to an entire OVM segment or to individual fields within the OVM segment by using field level access checking.

FSROOT | NOFSROOT

FSROOT(*file-system-root*)

Specifies the pathname for the file system root.

When you define the FSROOT pathname to RACF, it can contain 1-1023 characters, consist of any character, and be entered with or without single quotes. The following rules hold:

- If the pathname contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotes. For example, if the pathname is (123), you must enter FSROOT(' (123) ').
- If a single quote is intended to be part of the pathname, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

When entering the ALTUSER command, both uppercase and lowercase are accepted and maintained in the case in which they are entered. You should specify the fully-qualified pathname because RACF does not ensure that a valid pathname has been specified.

NOFSROOT

Specifies that you want to delete the FSROOT pathname from the OVM segment of the user's profile.

If you do not specify a value for FSROOT in the OVM segment, VM uses the value specified in the CP directory. If no value is specified in the CP directory, issue the OPENVM MOUNT command to mount the appropriate file system.

HOME | NOHOME

HOME(*initial-directory-name*)

sSpecifies the initial directory pathname. The initial directory is part of the file system and is the current working directory for the user's process when the user enters the OPENVM SHELL command.

When you define a HOME directory name to RACF, the name can contain 1-1023 characters, consist of any character, and be entered with or without single quotes. The following rules hold:

- If the pathname contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotes. For example, if the pathname is (123), you must enter HOME(' (123) ').
- If a single quote is intended to be part of the pathname, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

When entering the ALTUSER command, both uppercase and lowercase are accepted and maintained in the case in which they are entered. You should specify the fully-qualified pathname because RACF does not ensure that a valid pathname has been specified.

NOHOME

Specifies that you want to delete the initial directory pathname from the OVM segment of the user's profile.

If no value is specified for HOME in the OVM segment, VM uses the value specified in the CP directory. If no value is specified in the CP directory, VM sets the working directory for the user to "/", the root directory.

PROGRAM(*program-name*)

Specifies the PROGRAM pathname (z/OS UNIX shell program). This is the first program started when the OPENVM SHELL command is entered.

When you define a PROGRAM pathname to RACF, it can contain 1-1023 characters, consist of any character, and be entered with or without single quotes. The following rules apply:

- If the pathname contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotes. For example, if the pathname is (123), you must enter PROGRAM(' (123) ').
- If a single quote is intended to be part of the pathname, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

When entering the ALTUSER command, both uppercase and lowercase are accepted and maintained in the case in which they are entered. Specify the fully-qualified pathname. RACF does not ensure that a valid pathname is specified.

NOPROGRAM

Specifies that you want to delete the PROGRAM pathname from the OVM segment of the user's profile.

If no value is specified for PROGRAM in the OVM segment, VM uses the value specified in the CP directory. If no value is specified in the CP directory, VM gives control to the default z/OS UNIX shell program (/bin/sh) when a user issues the OPENVM SHELL command.

ALTUSER

UID(*user-identifier*)

Specifies the user identifier. The UID is a numeric value between 0 and 2 147 483 647.

Care should be taken in assigning 0 as the user identifier. UID 0 is considered a superuser, and a superuser passes all OpenExtensions VM security checks.

Notes:

1. RACF does not require the UID to be unique. You can assign the same value to multiple users, but this is not recommended because individual user control is lost. However, if you want a set of users to have exactly the same access to the OpenExtensions VM resources, you can assign the same UID to more than one user.
2. Exercise caution when changing the UID for a user.
 - The file-system might contain files that were created by the user, and thus contain the old UID as the file owner UID. Depending on the permission bits associated with the file, the user will probably lose access to those files.
 - If files already exist with an owner UID equal to the user's new UID value, the user will probably gain access to these files.
 - If another user is subsequently added with the old value as its UID, then the user might have access to the old files.
 - If you have an EXEC.*uid* profile in the VMPOSIX class for the old UID value, make sure you delete this profile and create another to reflect the new value.

NOUID

Specifies that you want to delete the user identifier from the OVM segment of the user's profile.

If NOUID is specified, the user is assigned the default UID of 4 294 967 295 (X'FFFFFFFF') and a LISTUSER for that user ID shows NONE for the UID.

NOOVM

Specifies that RACF delete the OVM segment from the user's profile.

OWNER(*userid or group-name*)

Specifies a RACF-defined user or group to be assigned as the new owner of the user's profile.

PASSWORD | NOPASSWORD

PASSWORD[(*password*)]

Specifies the user's temporary logon password. Use this command to specify a password for a user who has forgotten his/her password. Unless the NOEXPIRED operand is also specified, this password is set expired, thus requiring the user to change the password at next logon or job start. Note that the password syntax rules your installation defines using SETROPTS PASSWORD do not apply to this password, unless the NOEXPIRED operand is also included (see the NOEXPIRED operand).

If you specify PASSWORD without a value, the password is defaulted to the user's default group name. If you specify PASSWORD without a value and specify DFLTGRP, the default password is the user's old default group name.

Note: For Security Server Network Authentication Service support, this means the key is not generated for the default group. However, the default group continues to be used as the RACF password.

Note that if the installation is maintaining a password history, the password that was in effect prior to issuing this command is stored as part of this history.

NOPASSWORD

Specifies that the user does not need to supply an initial logon password when first entering the system if OIDCARD is also specified. If NOOIDCARD is specified, or the user ID has the NOOIDCARD attribute and you specify NOPASSWORD, you change the status of the user ID to protected. Protected user IDs cannot be used to enter the system by any means that requires a password to be specified, such as a TSO logon, CICS signon, batch job that specifies a password on the JOB statement. Therefore, user IDs that you assign to z/OS UNIX, UNIX daemons, started procedures, applications, servers or subsystems can be protected from being revoked when an incorrect password is entered. If the user attempts to enter the system with a password, the attempt fails. Note that the protected user ID is not revoked due to the failed password attempts even if the SETROPTS PASSWORD(REVOKE) option is in effect.

Determine which user IDs you wish to protect, ensuring that these user IDs will not be used in any circumstance where a password must be supplied. A protected user will have the PROTECTED attribute displayed in the output of the LISTUSER command. Protected users can be associated with started procedures defined in the STARTED class (preferred method) or in the started procedures table (ICHRIN03).

Notes:

1. Specifying PASSWORD, NOPASSWORD, OIDCARD, or NOOIDCARD for a protected user ID on a down-level system (OS/390 Version 2 Release 7 or earlier) should be avoided.
2. A protected user ID can still be revoked for failed password attempts from a down-level system (OS/390 Version 2 Release 7 or earlier).
3. For OS/390 Version 2 Release 8 and on, and for z/OS, RACF protected user IDs can be defined. These RACF user identities allow auditing and authorization, but are not intended for users (or other systems). Consequently, Security Server Network Authentication Service information such as a local *kerberos-principal-name* must not be defined for protected user IDs, and these user IDs must not be used for Network Authentication Service authentication, since these authentication failures can result in user revocation.

PROXY | NOPROXY

PROXY

Specifies information which the z/OS LDAP Server will use when acting as a proxy on behalf of a requester. The R_proxyserv (IRRSPY00) SAF callable service will attempt to retrieve this information when it is not explicitly supplied via invocation parameters. Applications or other services which use the R_proxyserv callable service, such as IBM Policy Director Authorization Services for z/OS and OS/390, may instruct their invokers to define PROXY segment information.

LDAPHOST(*ldap_url*)

Specifies the URL of the LDAP server which the z/OS LDAP Server will contact when acting as a proxy on behalf of a requester. An LDAP URL has a format such as `ldap://123.45.6:389` or `ldaps://123.45.6:636`, where `ldaps` indicates that an SSL connection is desired for a higher level of security. LDAP will also allow you to specify the host name

ALTUSER

portion of the URL using either the text form (BIGHOST.POK.IBM.COM) or the dotted decimal address (123.45.6). The port number is appended to the host name, separated by a colon ':' (X'7A'). See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP URLs and how to enable LDAP servers for SSL connections.

The LDAP URL that you define to RACF can consist of 10—1023 characters. A valid URL must start with either `ldap://` or `ldaps://`. RACF will allow any characters to be entered for the remaining portion of the URL, but you should ensure that the URL conforms to TCP/IP conventions. For example, parentheses, commas, blanks, semicolons, and single quotes are not typically allowed in a host name. The LDAP URL can be entered with or without single quotes, however, in both cases, it will be folded to uppercase.

RACF does not ensure that a valid LDAP URL has been specified.

NOLDAPHOST

Deletes the URL of the LDAP server which the z/OS LDAP Server will contact when acting as a proxy on behalf of a requester.

BINDDN(*bind_distinguished_name*)

Specifies the distinguished name (DN) which the z/OS LDAP Server will use when acting as a proxy on behalf of a requester. This DN will be used in conjunction with the BIND password, if the z/OS LDAP Server needs to supply an administrator or user identity to BIND with another LDAP Server. A DN is made up of attribute value pairs, separated by commas. For example:

```
cn=Ben Gray,ou=editing,o=New York Times,c=US
cn=Lucille White,ou=editing,o=New York Times,c=US
cn=Tom Brown,ou=reporting,o=New York Times,c=US
```

See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP DNs.

When you define a BIND DN to RACF, it can contain 1-1023 characters. The BIND DN can consist of any characters and can be entered with or without single quotes. The following rules apply:

- If parentheses, commas, blanks, or semicolons are to be entered as part of the BIND DN, the character string must be enclosed in single quotes.
- If a single quote is intended to be part of the BIND DN, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered. See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP distinguished names.

If you issue the ALTUSER command as a RACF operator command and you specify the BIND DN in lowercase, you must include the BIND DN within single quotations.

RACF does not ensure that a valid BIND DN has been specified.

NOBINDDN

Deletes the distinguished name (DN) used by the z/OS LDAP server when acting as a proxy on behalf of a requester.

BINDPW

Specifies the password which the z/OS LDAP Server will use when acting as a proxy on behalf of a requester.

When you define a BIND password to RACF, it can contain 1-128 characters. The BIND password can consist of any characters (see exception below) and can be entered with or without single quotes. The following rules apply:

- The BIND password can not start with a left curly brace '{' (X'8B').
- If parentheses, commas, blanks, or semicolons are to be entered as part of the BIND password, the character string must be enclosed in single quotes.
- If a single quote is intended to be part of the BIND password, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered. See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP passwords.

If you issue the ALTUSER command as a RACF operator command and you specify the BIND password in lowercase, you must include the BIND password within single quotations.

RACF does not ensure that a valid BIND password has been specified.

Attention:

- When the command is issued from ISPF, the TSO command buffer (including possible BINDPW password data) is written to the ISPLOG data set. As a result, you should not issue this command from ISPF or you must control the ISPLOG data set carefully.
- When the command is issued as a RACF operator command, the command and the possible BINDPW password data is written to the system log. Therefore, use of RALTER as a RACF operator command should either be controlled or you should issue the command as a TSO command.

NOBINDPW

Deletes the password used by the z/OS LDAP server when acting as a proxy on behalf of a requester.

NOPROXY

Deletes LDAP proxy information.

RESTRICTED | NORESTRICTED**RESTRICTED**

Specifies that global access checking is bypassed when resource access checking is performed for the user, and neither ID(*) on the access list nor the UACC will allow access. The RESTRICTED.FILESYS.ACCESS profile in the UNIXPRIV class can also be used to bypass the z/OS UNIX 'other' permission bits during file access checking for RESTRICTED users.

ALTUSER

Note: If your installation has profiles defined in the PROGRAM class, and the user ID with the RESTRICTED attribute needs to load programs covered by one or more of these profiles, the user id must be put on the access list with EXECUTE or READ authority.

NORESTRICTED

Specifies that the user does not have the RESTRICTED attribute and access checking is performed the standard way including global access checking, ID(*), the UACC, and the z/OS UNIX 'other' permission bits as appropriate.

RESUME[(date)]

Specifies that the user is to be allowed to access the system again. You normally use RESUME to restore access to the system that has been prevented by a prior REVOKE.

If you specify a date, RACF prevents the user from accessing the system until the date you specify. The date must be a future date; if it is not, you are prompted to provide a future date.

During the time between when you specify the RESUME and the date when the RESUME takes effect, the RESUME is called a pending RESUME. You specify a date in the form *mm/dd/yy*, and you need not specify leading zeros; specifying 9/1/94 is the same as specifying 09/01/94. RACF interprets dates as 20yy when yy is less than 71, and 19yy is 71 or higher. So, 09/01/94 would be in the year 1994, and 09/01/14 would be in the year 2014.

If you specify RESUME without a date, the RESUME takes effect immediately. Specifying RESUME without a date overrides any pending RESUME and any pending REVOKE.

When no REVOKE or pending REVOKE is in effect for the user, RACF ignores the RESUME operand.

Notes:

1. If you use the ALTUSER command to issue a REVOKE for a user, you must use the ALTUSER command to issue the corresponding RESUME. Issuing RESUME on the CONNECT command does not restore access revoked on the ALTUSER command.
2. If you specify both REVOKE(*date*) and RESUME(*date*), RACF acts on them in date order. For example, if you specify RESUME(8/19/94) and REVOKE(8/5/94), RACF prevents the user from accessing the system from August 5, 1994, to August 18, 1994. On August 19, the user can again access the system.

If a user is already revoked and you specify RESUME(8/5/94) and REVOKE(8/19/94), RACF allows the user to access the system from August 5, 1994, to August 18, 1994. On August 19, RACF prevents the user from accessing the system.
3. If RACF detects a conflict between REVOKE and RESUME (for example, you specify both without a date), RACF uses REVOKE. If you specify, without a date, only REVOKE or only RESUME, RACF clears both date fields. When RACF revokes a user because of inactivity or invalid password attempts, RACF also clears both date fields.

REVOKE[(date)]

Specifies that RACF is to prevent the user from accessing the system. The user's profile is not deleted from the RACF database, and the user's data sets are not deleted from the RACF data set.

If you specify the date, RACF prevents the user from accessing the system, starting on the date you specify. The date must be a future date; if it is not, you are prompted to provide a future date.

During any time between when you specify the REVOKE and the date when the REVOKE takes effect, the REVOKE is called a pending revoke.

You specify a date in the form *mm/dd/yy*, and you need not specify leading zeros; specifying 9/1/94 is the same as specifying 09/01/94. RACF interprets dates as 20yy when yy is less than 71, and 19yy when yy is 71 or higher. So, 09/01/94 would be in the year 1994, and 09/01/14 would be in the year 2014.

If you specify REVOKE without a date, the REVOKE takes effect the next time the user tries to log onto the system. Specifying REVOKE without a date overrides any pending REVOKE. Note that RESUME works the same way, as specifying RESUME without a date will also override any pending REVOKE.

When a REVOKE is already in effect for the user, RACF ignores the REVOKE operand and issues a message.

Notes:

1. Specifying REVOKE on the ALTUSER command overrides RESUME on the CONNECT command.
2. If you specify both REVOKE(*date*) and RESUME(*date*), RACF acts on them in date order. For example, if you specify RESUME(8/19/94) and REVOKE(8/5/94), RACF prevents the user from accessing the system from August 5, 1994, to August 18, 1994. On August 19, the user can again access the system.

If a user is already revoked and you specify RESUME(8/5/94) and REVOKE(8/19/94), RACF allows the user to access the system from August 5, 1994, to August 18, 1994. On August 19, RACF prevents the user from accessing the system.

3. If RACF detects a conflict between REVOKE and RESUME (for example, you specify both without a date), RACF uses REVOKE. If you specify, without a date, only REVOKE or only RESUME, RACF clears both date fields. When RACF revokes a user because of inactivity or invalid password attempts, RACF also clears both date fields.

SECLABEL | NOSECLABEL

SECLABEL(*seclabel-name*)

Specifies the user's default security label where *seclabel-name* is an installation-defined security label that represents an association between a particular security level and a set of zero or more security categories.

A security label corresponds to a particular security level (such as CONFIDENTIAL) with a set of zero or more security categories (such as PAYROLL or PERSONNEL).

When the SECLABEL class is not active, RACF ignores this operand. When no member of the SECLABEL profile exists for *seclabel-name*, you are prompted to provide a valid security label name.

NOSECLABEL

Specifies that the ALTUSER command is to delete any security label contained in the user profile. The user no longer has access to any resource that requires a requester to have a certain security label.

SECLEVEL | NOSECLEVEL

ALTUSER

SECLEVEL(*secllevel-name*)

Specifies the user's security level, where *secllevel-name* is an installation-defined name that must be a member of the SECLEVEL profile in the SECDATA class. The security level name that you specify corresponds to the number of the minimum security level that a user must have to access the resource.

When you specify SECLEVEL and the SECDATA class is active, RACF adds security level access checking to its other authorization checking. If global access checking does not grant access, RACF compares the security level allowed in the user profile with the security level required in the resource profile. If the security level in the user profile is less than the security level in the resource profile, RACF denies the access. If the security level in the user profile is equal to or greater than the security level in the resource profile, RACF continues with other authorization checking.

Note: RACF does not perform security level checking for a started task or user that has the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF started procedures table or STARTED class, or to other users by customer-supplied RACF exits.

When the SECDATA class is not active, RACF ignores this operand. When the SECLEVEL profile does not include a member for *secllevel-name*, you are prompted to provide a valid security level name.

NOSPECIAL

Specifies that the ALTUSER command is to delete any security level contained in the user profile. The user no longer has access to any resource that requires a requester to have a certain security level.

SPECIAL | NOSPECIAL

SPECIAL

Specifies that the user is to be allowed to issue all RACF commands with all operands except the operands that require the AUDITOR attribute. SPECIAL specified on the ALTUSER command overrides NOSPECIAL specified on the CONNECT command.

You must have the SPECIAL attribute to use the SPECIAL operand.

NOSPECIAL

Specifies that the user no longer has the SPECIAL attribute.

You must have the SPECIAL attribute to use the NOSPECIAL operand.

TSO | NOTSO

TSO

Specifies that when you change the profile of a TSO user, you can enter any of the following suboperands to add or change default TSO logon information for that user. Each suboperand defines information that RACF stores in a field within the TSO segment of the user's profile.

You can control access to an entire TSO segment or to individual fields within the TSO segment by using field-level access checking. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

ACCTNUM(*account-number*)

Specifies the user's default TSO account number when logging on from

the TSO/E logon panel. The account number you specify must be defined as a profile in the ACCTNUM general resource class, and the user must be granted READ access to the profile. Otherwise, the user cannot log on to TSO using the specified account number.

Account numbers can consist of any characters, and can be entered with or without single quotes. The following rules apply:

- If parentheses, commas, blanks, and semicolons are to be entered as part of the account number, the character string must be enclosed in single quotes. For example, if the account number is (123), you must enter ACCTNUM(' (123) ').
- If a single quote is intended to be part of the account number, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

A user can change an account number, or specify an account number if one has not been specified, using the TSO/E logon panel. RACF checks the user's authorization to the specified account number. If the user is authorized to use the account number, RACF stores the account number in the TSO segment of the user's profile, and TSO/E uses it as a default value the next time the user logs on to TSO/E. Otherwise, RACF denies the use of the account number.

Note: When you define an account number on TSO, you can specify 1-40 characters. When you define a TSO account number to RACF, you can specify only 1-39 characters.

NOACCTNUM

Specifies that you want to delete the user's default account number. If you delete the default account number from a user's profile, RACF uses a default value consistent with current TSO defaults when the user logs on to TSO.

COMMAND(*command-issued-at-logon*)

Specifies the command to be run during TSO/E logon. TSO/E uses this field to prime the COMMAND field of the logon panel. The command value can contain 1-80 characters and consist of any characters. You can enter the value with or without single quotes depending on the following rules:

- If the command value contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotes. For example, if the command value is (123), you must enter COMMAND(' (123) ').
- If a single quote is intended to be part of the command value, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered. A user can change the command value, or specify a command if one has not been specified, using the TSO/E logon panel.

Note: It is recommended that you use this command for a user who is logged off. If you change the command value for a

ALTUSER

currently logged-on user ID, the change is overwritten by the TSO/E logoff command processor when the user ID is logged off.

NOCOMMAND

Deletes any COMMAND data that was previously saved in the RACF database for this user ID.

Note: When you delete this field for a currently logged-on user ID, the field is overwritten by the TSO/E logoff command processor when the user ID is logged off.

DEST | NODEST

DEST(*destination-id*)

Specifies the default destination to which the user can route dynamically allocated SYSOUT data sets. The specified value must be 1-7 alphanumeric characters, beginning with an alphabetic or national character.

NODEST

Specifies that you want to remove any default destination information for this user. Without explicit action by the user to route SYSOUT, the SYSOUT for this user is printed at your system default print location.

HOLDCLASS(*hold-class*) | **NOHOLDCLASS**

HOLDCLASS(*hold-class*)

Specifies the user's default hold class. The specified value must be 1 alphanumeric character, excluding national characters.

If you specify the TSO operand on the ALTUSER command but do not specify a value for HOLDCLASS, RACF uses a default value consistent with current TSO defaults.

NOHOLDCLASS

Specifies that you want to delete the default hold class from the TSO segment of the user's profile. If you delete the default hold class from a user's profile, RACF uses a default value consistent with current TSO defaults when the user logs onto TSO.

JOBCLASS(*job-class*) | **NOJOBCLASS**

JOBCLASS(*job-class*)

Specifies the user's default job class. The specified value must be 1 alphanumeric character, excluding national characters.

If you specify the TSO operand on the ALTUSER command but do not specify a value for JOBCLASS, RACF uses a default value consistent with current TSO defaults.

NOJOBCLASS

Specifies that you want to delete the default job class from the TSO segment of the user's profile. If you delete the default job class from a user's profile, RACF uses a default value consistent with current TSO defaults when the user logs on to TSO.

MAXSIZE(*maximum-region-size*) | **NOMAXSIZE**

MAXSIZE(*maximum-region-size*)

Specifies the maximum region size that the user can request at logon. The maximum region size is the number of 1024-byte units

of virtual storage that TSO can create for the user's private address space. The specified value must be an integer in the range of 0 through 65535 if the database is shared with any MVS/370 systems, or 0 through 2096128 if the database is not shared with any MVS/370 systems.

If you specify the TSO operand on the ALTUSER command but do not specify a value for MAXSIZE, or specify MAXSIZE(0), RACF uses a default value consistent with current TSO defaults.

If values are specified for both MAXSIZE and SIZE and SIZE is greater than MAXSIZE, RACF sets SIZE equal to MAXSIZE. If a value is specified for only SIZE or MAXSIZE and SIZE is greater than MAXSIZE, the operand is ignored.

If your installation is sharing a database between MVS and VM, the administrator is allowed to specify the largest possible value. If your installation is sharing a database between MVS systems, and one of the systems sharing the database is an MVS/370 system, the administrator must not specify a value greater than 65535.

NOMAXSIZE

Specifies that you want to delete the maximum region size from the TSO segment of the user's profile. If you delete the maximum region size from a user's profile, RACF uses a default value consistent with current TSO defaults when the user logs on to TSO.

MSGCLASS(*message-class*) | NOMSGCLASS

MSGCLASS(*message-class*)

Specifies the user's default message class. The specified value must be 1 alphanumeric character, excluding national characters.

If you specify the TSO operand on the ALTUSER command but do not specify a value for MSGCLASS, RACF uses a default value consistent with current TSO defaults.

NOMSGCLASS

Specifies that you want to delete the default message class from the TSO segment of the user's profile. If you delete the default message class from a user's profile, RACF uses a default value consistent with current TSO defaults when the user logs on to TSO.

PROC | NOPROC

PROC(*logon-procedure-name*)

Specifies the name of the user's default logon procedure when logging on through the TSO/E logon panel. The name you specify must be 1-8 alphanumeric characters and begin with an alphabetic character. The name must also be defined as a profile in the TSOPROC general resource class, and the user must be granted READ access to the profile. Otherwise, the user cannot log on to TSO using the specified logon procedure.

A user can change a logon procedure, or specify a logon procedure if one has not been specified, using the TSO/E logon panel. RACF checks the user's authorization to the specified logon procedure. If the user is authorized to use the logon procedure, RACF stores the name of the procedure in the TSO segment of the user's profile, and TSO/E uses it as a default value the next time the user logs on to TSO/E. Otherwise, RACF denies the use of the logon procedure.

ALTUSER

NOPROC

Specifies that you want to delete the default logon procedure from the TSO segment of the user's profile. If you delete the default logon procedure from a user's profile, RACF uses a default value consistent with current TSO defaults when the user logs on to TSO.

SECLABEL | NOSECLABEL

SECLABEL(*security-label*)

Specifies the user's security label if the user specifies one on the TSO logon panel.

NOSECLABEL

Specifies that you want to delete the security label from the TSO segment of the user's profile. If you delete the security label from a user's TSO segment, RACF uses the security label in the user's profile the next time the user logs on to TSO.

SIZE | NOSIZE

SIZE(*default-region-size*)

Specifies the minimum region size if the user does not request a region size at logon. The default region size is the number of 1024-byte units of virtual storage available in the user's private address space at logon. The specified value must be an integer in the range of 0 through 65535 if the database is shared with any MVS/370 systems, or 0 through 2096128 if the database is not shared with any MVS/370 systems.

A user can change a minimum region size, or specify a minimum region size if one has not been specified, using the TSO/E logon panel. RACF stores this value in the TSO segment of the user's profile, and TSO/E uses it as a default value the next time the user logs on to TSO/E.

If values are specified for both MAXSIZE and SIZE and SIZE is greater than MAXSIZE, RACF sets SIZE equal to MAXSIZE. If a value is specified for only SIZE or MAXSIZE and SIZE is greater than MAXSIZE, the operand is ignored.

If your installation is sharing a database between MVS and VM, the administrator is allowed to specify the largest possible value. If your installation is sharing a database between MVS systems, and one of the systems sharing the database is an MVS/370 system, the administrator must not specify a value greater than 65535.

NOSIZE

Specifies that you want to delete the default minimum region size from the TSO segment of the user's profile. If you delete the default minimum region size from a user's profile, RACF uses a default value consistent with current TSO defaults when the user logs on to TSO.

SYSOUTCLASS(*sysout-class*) | NOSYSOUTCLASS

SYSOUTCLASS(*sysout-class*)

Specifies the user's default SYSOUT class. The specified value must be 1 alphanumeric character, excluding national characters.

If you specify the TSO operand on the ALTUSER command but do not specify a value for SYSOUTCLASS, RACF uses a default value consistent with current TSO defaults.

NOSYSOUTCLASS

Specifies that you want to delete the default SYSOUT class from the TSO segment of the user's profile. If you delete the default SYSOUT class from a user's profile, RACF uses a default value consistent with current TSO defaults when the user logs on to TSO.

UNIT | NOUNIT**UNIT**(*unit-name*)

Specifies the default name of a device or group of devices that a procedure uses for allocations. The specified value must be 1-8-alphanumeric characters.

NOUNIT

Specifies that you want to delete the default name of a device or group of devices that a procedure uses for allocations from the TSO segment of the user's profile. If you delete this name from a user's profile, RACF uses a default value consistent with current TSO defaults when the user logs on to TSO.

USERDATA | NOUSERDATA**USERDATA**(*user-data*)

Specifies optional installation data defined for the user. The specified value must be 4 EBCDIC characters; valid characters are 0 through 9 and A through F.

Note: When you change this value for a currently logged-on user ID, the change is overwritten by the TSO logoff command processor when the user ID is logged off.

NOUSERDATA

Specifies that you want to delete the installation data previously defined for a user.

NOTSO

Specifies that you are revoking a user's authority to use TSO. RACF deletes TSO logon information from the RACF database for the specified user. However, if the user ID is currently logged on, when the user issues the LOGOFF command the TSO logoff processor restores the TSO segment with default values (except for the USERDATA field which is set to the user's current value). To prevent the TSO segment from being restored, the user ID should be logged off before issuing the ALTUSER NOTSO command.

When you specify NOTSO, the result is the same as if you issue the TSO ACCOUNT command with the DELETE subcommand.

UACC(*access-authority*)

Specifies the new default universal access authority for all new resources the user defines while connected to the specified default group. The universal access authorities are ALTER, CONTROL, UPDATE, READ, and NONE. (RACF does not accept EXECUTE access authority with the ALTUSER command.) If you specify UACC without a value, RACF ignores the operand.

This option is group-related. If the user is connected to other groups, the user can have a different default universal access authority in each group.

Note: When an MVS user (who has the ADSP attribute or specifies the PROTECT parameter on a JCL DD statement) enters the system using the group specified in the GROUP operand as the current connect

ALTUSER

group, RACF assigns this default universal access authority to any data set or tape volume RACF profiles the user defines.

UAUDIT | NOUAUDIT

UAUDIT

Specifies that RACF is to log all RACROUTE REQUEST=AUTH and RACROUTE REQUEST=FASTAUTH services that are eligible for logging, and all RACROUTE REQUEST=DEFINE services issued for the user, and all RACF commands (except SEARCH, LISTDSD, LISTGRP, LISTUSER, and RLIST) issued by the user. (When you change a user profile and omit both UAUDIT and NOUAUDIT, the default is NOUAUDIT.)

You must have the AUDITOR attribute, or the user profile must be within the scope of a group in which you have the group-AUDITOR attribute, in order to enter the UAUDIT operand.

If an unauthorized user specifies UAUDIT on the ALTUSER command, none of the operands on the command is processed. RACF issues ICH21005I NOT AUTHORIZED TO SPECIFY UAUDIT, OPERAND IGNORED. The System Action states "RACF ignores the operand and continues processing with the next operand". RACF verifies other operands, but does not process any of them. For more information, see *z/OS Security Server RACF Messages and Codes*.

NOUAUDIT

Specifies that no UAUDIT logging is to be performed. This operand does not override any other auditing options (for example, CMDVIOL specified on SETROPTS) that might be in effect.

You must have the AUDITOR attribute, or the user profile must be within the scope of a group in which you have the group-AUDITOR attribute, to enter the NOUAUDIT operand.

WHEN([DAYS(*day-info*)] [TIME(*time-info*)])

Specifies the days of the week and the hours in the day when the user is allowed to access the system from a terminal. The day-of-week and time restrictions apply only when a user logs on to the system; that is, RACF does not force the user off the system if the end-time occurs while the user is logged on. Also, the day-of-week and time restrictions do not apply to batch jobs; the user can submit a batch job on any day and at any time.

If you specify the WHEN operand, you can restrict the user's access to the system to certain days of the week and to a certain time period within each day. For example, you can restrict a user's access to any one of the following:

- From 9:00 a.m. to 5:00 p.m. (0900:1700); this would be a daily restriction since days were not also specified.
- Monday through Friday; this restriction applies for all 24 hours of Monday, Tuesday, Wednesday, Thursday, and Friday.
- Monday through Friday from 9:00 a.m. to 5:00 p.m. (0900:1700)

Note: You cannot specify more than one combination of days and times, even through multiple ALTUSER commands. For example, if you specify:

```
ALTUSER user_ID WHEN(DAYS(MONDAY TUESDAY) TIME(0100:0500))
ALTUSER user_ID WHEN(DAYS(THURSDAY) TIME(0200:0500))
```

the result is that *user_ID* is allowed to access the system only on Thursday from 2:00 to 5:00; the preceding DAYS (MONDAY TUESDAY) and TIME (0100:0500) operands are overwritten.

To allow a user to access the system only on certain days, specify `DAYS(day-info)`, where *day-info* can be any one of the following:

ANYDAY

Specifies that the user can access the system on any day.

WEEKDAYS

Specifies that the user can access the system only on weekdays (Monday through Friday).

day ...

Specifies that the user can access the system only on the days specified, where *day* can be MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, or SUNDAY, and you can specify the days in any order.

To allow a user to access the system only during a certain time period of each day, specify `TIME(time-info)`, where *time-info* can be any one of the following:

ANYTIME

Specifies that the user can access the system at any time.

start-time:end-time

Specifies that the user can access the system only during the specified time period. The format of both *start-time* and *end-time* is *hhmm*, where *hh* is the hour in 24-hour notation (00 through 23) and *mm* is the minutes (00 through 59). Note that 0000 is not a valid time value.

If *start-time* is greater than *end-time*, the interval spans midnight and extends into the following day.

If you omit `DAYS` and specify `TIME`, the time restriction applies to any day-of-week restriction already indicated in the profile. If you omit `TIME` and specify `DAYS`, the day restriction applies to the time restriction already indicated in the profile. If you specify both `DAYS` and `TIME`, the user can access the system only during the specified time period and only on the specified days.

If you omit both `DAYS` and `TIME`, the time and day restriction remains as it was in the profile.

WORKATTR | NOWORKATTR**WORKATTR**

Specifies the user-specific attributes of a unit of work.

z/OS elements or features such as APPC, WLM, and z/OS UNIX System Services might use the `WORKATTR` segment.

These operands are used by APPC/MVS for SYSOUT created by APPC transactions.

WAACNT(*account-number*) | NOWAACNT

Specifies an account number for APPC/MVS processing.

You can specify a maximum of 255 EBCDIC characters. Use the following rules when entering a value for this field:

- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotes. For example, if the data is (123), you must enter `WAACNT '(123)'`.

- If a single quote is intended to be part of the data, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

NOWAACNT deletes the account number from the user profile.

WAADDR n (address-line- n) | NOWAADDR n

Where n can be from 1 to 4, *address-line- n* specifies other address lines for SYSOUT delivery. For each line of the address you can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase are accepted and maintained in the case in which they are entered.

Use the following rules when entering a value for this field:

- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotes. For example, if the data is (123), you must enter WAADDR(' (123)').
- If a single quote is intended to be part of the data, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

NOWAADDR deletes address line n from the user profile.

WABLDG(building) | NOWABLDG

Specifies the building that SYSOUT information is to be delivered to.

You can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase are accepted and maintained in the case in which they are entered.

Use the following rules when entering a value for this field:

- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotes. For example, if the data is (123), you must enter WABLDG(' (123)').
- If a single quote is intended to be part of the data, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

NOWABLDG deletes the building from the profile.

WADEPT(department) | NOWADEPT

Specifies the department that SYSOUT information is to be delivered to.

You can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase are accepted and maintained in the case in which they are entered.

Use the following rules when entering a value for this field:

- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotes. For example, if the data is (123), you must enter WADEPT(' (123)').
- If a single quote is intended to be part of the data, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

NOWADEPT deletes the department from the profile.

WANAME(name) | NOWANAME

Specifies the name of the user SYSOUT information is to be delivered to.

You can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase are accepted and maintained in the case in which they are entered.

Use the following rules when entering a value for this field:

- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotes. For example, if the data is (123), you must enter WANAME(' (123) ').
- If a single quote is intended to be part of the data, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

NOWANAME deletes the name from the profile.

WAROOM(room) | NOWAROOM

Specifies the room SYSOUT information is to be delivered to.

You can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase are accepted and maintained in the case in which they are entered.

Use the following rules when entering a value for this field:

- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotes. For example, if the data is (123), you must enter WAROOM(' (123) ').
- If a single quote is intended to be part of the data, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

NOWAROOM deletes the room from the profile.

NOWORKATTR

Specifies that you want to delete the work attributes previously defined for a user.

Examples

Table 14. ALTUSER Examples

Example 1	Operation	User IA0 wants to alter the level of group authority from USE to CREATE for user DAF0 in the user's (DAF0's) default group so user DAF0 can define generic profiles for data sets in group RESEARCH.
	Known	User IA0 is the owner of user DAF0 and has JOIN authority in the group RESEARCH.
Example 2		The default group for user DAF0 is RESEARCH.
		User IA0 wants to issue the command as a RACF TSO command.
	Command	ALTUSER DAF0 AUTHORITY(CREATE)
	Defaults	GROUP(RESEARCH)
	Operation	User CD0 wants to correct his name and change his default group to PAYROLL.
	Known	The default group for user CD0 is RESEARCH.
		User CD0 has USE authority in the group PAYROLL.
		User CD0 wants to issue the command as a RACF TSO command.
	Command	ALTUSER CD0 NAME(CDAVIS) DFLTGRP(PAYROLL)
	Defaults	None

ALTUSER

Table 14. ALTUSER Examples (continued)

Example 3	<i>Operation</i>	User IA0 wants to add the FINANCIAL category and the CONFIDENTIAL security level to user ESH25's profile and restrict the user's access to the system to weekdays from 8:00 a.m. to 8:00 p.m.
	<i>Known</i>	User IA0 is connected to group PAYROLL with the group-SPECIAL attribute. Group PAYROLL is user ESH25's default group. User IA0's profile includes the FINANCIAL category and the CONFIDENTIAL security level. The FINANCIAL category and the CONFIDENTIAL security level have been defined to RACF.
	<i>Command</i>	User IA0 wants to issue the command as a RACF TSO command. ALTUSER ESH25 ADDCATEGORY(FINANCIAL) SECLEVEL(CONFIDENTIAL) WHEN(DAYS(WEEKDAYS)TIME(0800:2000))
Example 4	<i>Defaults</i>	None
	<i>Operation</i>	User RADM02 wants to revoke the user ID of an employee, user D5819, who will be on vacation for three weeks, starting on August 5, 1994. User RADM02 wants to direct the command to run at the local node under the authority of user HICKS and prohibit the command from being automatically directed to other nodes.
	<i>Known</i>	Users RADM02 and HICKS have the SPECIAL attribute. Today's date is August 3, 1994. User RADM02 wants to issue the command as a RACF TSO command. Users RADM02 and HICKS have an already established user ID association.
Example 5	<i>Command</i>	ALTUSER D5819 REVOKE(8/5/94) RESUME(8/26/94) ONLYAT(.HICKS)
	<i>Results</i>	The command is only processed on the local node and not automatically directed to any other nodes in the RRSF configuration.
	<i>Operation</i>	User RGB01 wants to remove all class authorities and the AUDITOR attribute from USER1, and wants to audit all activity by user USER1.
Example 6	<i>Known</i>	User RGB01 has the SPECIAL and AUDITOR attributes. User USER1 is an existing user. User RGB01 wants to issue the command as a RACF TSO command.
	<i>Command</i>	ALTUSER USER1 NOCLAUTH(USER TERMINAL) NOAUDITOR UAUDIT
	<i>Defaults</i>	None
Example 6	<i>Operation</i>	User RADMIN wants to change the installation-defined information contained in the SJR1 user ID entry, and delete the model name information.
	<i>Known</i>	User RADMIN is the owner of user ID SJR1. User RADMIN wants to issue the command as a RACF TSO command.
	<i>Command</i>	ALTUSER SJR1 DATA('RESOURCE USAGE ADMINISTRATOR NAME TOM P.') NOMODEL
	<i>Defaults</i>	None

Table 14. ALTUSER Examples (continued)

Example 7

Operation User VROGERS wants to change default TSO logon information for user BNORTH. User BNORTH requires the following changes:

- A new TSO account number, 12345
- A new TSO logon procedure, LPROC12
- A new SYSOUT data set destination, BL2030
- A new SYSOUT class, Z
- A new maximum region size, 18000.

Known

- User VROGERS has the SPECIAL attribute.
- User BNORTH has been defined to RACF with authority to use TSO.
- 12345 has been defined to RACF as a profile in the ACCTNUM general resource class, and user BNORTH has been given READ access to this profile.
- LPROC12 has been defined to RACF as a profile in the TSOPROC general resource class, and user BNORTH has been given READ access to this profile.
- User VROGERS wants to issue the command as a RACF TSO command.

Command ALTUSER BNORTH TSO(ACCTNUM(12345) PROC(LPROC12) DEST(BL2030) SYSOUTCLASS(Z) MAXSIZE(18000))

Defaults None

Example 8

Operation User MIKEM wants to make the following changes to the profile for user MARTIN:

- Change the default DFP management class to MGMT617
- Change the default DFP storage class to STOR533
- Delete the default DFP data application.

Known

- User MIKEM has the SPECIAL attribute.
- User MARTIN has been defined to RACF, and MARTIN's user profile contains a DFP segment.
- MGMT617 has been defined to RACF as a profile in the MGMTCLAS general resource class, and user MARTIN has been given READ access to this profile.
- STOR533 has been defined to RACF as a profile in the STORCLAS general resource class, and user MARTIN has been given READ access to this profile.
- User MIKEM wants to issue the command as a RACF TSO command.

Command ALTUSER MARTIN DFP(MGMTCLAS(MGMT617) STORCLAS(STOR533) NODATAAPPL))

Defaults None

Example 9

Operation A user with SPECIAL authority wants to make existing z/OS UNIX System Services user CSMITH a superuser and delete PROGRAM from CSMITH's profile so that the default z/OS UNIX shell program is used when CSMITH enters the TSO/E command OMVS.

Known User CSMITH is already defined to OMVS. The user with SPECIAL authority wants to issue the command as a RACF TSO command.

Command ALTUSER CSMITH OMVS(UID(0) NOPROGRAM)

Defaults None

Example 10

Operation A user with SPECIAL authority wants to make existing z/OS UNIX System Services DCE user, CSMITH, a z/OS UNIX System Services superuser and change the HOMECCELL name to ../../hottie.scarol.ibm.com.

Known The DCE UUID for the ../../hottie.scarol.ibm.com cell is 003456ab-ecb7-7de3-ebda-95531ed63dae.

Command ALTUSER CSMITH OMVS(UID(0)) + DCE(HOMECCELL('../../hottie.scarol.ibm.com') + HOMEUUID(003456ab-ecb7-7de3-ebda-95531ed63dae))

Defaults None

ALTUSER

Table 14. ALTUSER Examples (continued)

Example 11

Operation A help desk consultant wants to reset a user's password.

- Known*
- The consultant is authorized to reset passwords
 - The consultant's RACF User ID (or RACF group associated with the help desk consultant's user ID) has been permitted by the security administrator with **READ** access to the RACF FACILITY class profile **IRR.PASSWORD.RESET**.
 - The help desk consultant is resetting user JIMBOB's password.

Command ALTUSER JIMBOB PASSWORD(TEMP012X)

Defaults EXPIRED

Example 12

Operation A help desk consultant wants to reset an application's password.

Known A help desk consultant has been authorized to reset passwords. The consultant's RACF user ID (or the RACF group associated with the consultant's user ID) has been permitted by the security administrator with UPDATE access to the RACF FACILITY class profile IRR.PASSWORD.RESET.

In this example, at the request of operations personnel, the consultant is resetting the user ID associated with an application called CUSTAPP.

The consultant uses the NOEXPIRED operand so the application user ID (CUSTAPP in this example) does not need to change the password when it is logged on.

To reset the application's password, the consultant enters:

Command ALTUSER CUSTAPP PASSWORD(STBR01R) NOEXPIRED

Note: the password value STBR01R must satisfy the password quality rules enforced by both SETROPTS and ICHPWX01.

Defaults None

Example 13

Operation User RACFADM with SPECIAL or UPDATE authority requests the alteration of a RACF user to add Lotus Notes information and to delete the NDS segment from the user's profile.

Known User RACFADM has SPECIAL authority or UPDATE authority to the desired field within the segment.

Command ALTUSER PCUSER2 LNOTES(SNAME(B.B.SMITH)) NONDS

Defaults None

Example 14

Operation User RACFADM with SPECIAL authority adds the user IDs PUBLIC, RACFU00, and USER04. The user ID PUBLIC is then altered and is assigned RESTRICTED access.

Known User RACFADM has SPECIAL authority.

Command ADDUSER (PUBLIC RACFU00 USER004)
ALTUSER PUBLIC RESTRICTED
ADDSD 'RACFU00.*' UACC(READ)

Defaults RACFU00, USER004, and PUBLIC have NORESTRICTED access by default.

Example 15

Operation An existing user, whose RACF user profile is RONTOMS, is defining a Security Server Network Authentication Service account within the local realm. MAXTKTLFE is not specified, so the value specified on the definition of the local realm KERBDFLT in the REALM class is used.

Known User RONTOMS wants to alter his user profile in order to add Security Server Network Authentication Service information.

Command ALTUSER RONTOMS KERB(KERBNAME('KerberizedUser'))
PASSWORD(BUNG21R) NOEXPIRED

Defaults None

Example 16

Operation User RACFADMIN issues a command to delete the profile that references the EIM domain in the LDAPBIND class for user MRSERVER.

Known The profile in the LDAPBIND class that defines the EIM LDAP values is no longer required for EIM processing

Command ALTUSER MRSERVER EIM(NOLDAPPROF)

Defaults None

CONNECT (Connect User to Group)

Purpose

Use the CONNECT command to connect a user to a group, modify a user's connection to a group, or assign the group-related user attributes. If you are creating a connection, defaults are available as stated for each operand. If you are modifying an existing connection, no defaults apply.

Note: RACF interprets dates with 2 digit years in the following way. YY represents the 2 digit year.

IF 70 < YY <= 99 THEN

The date is interpreted as 19YY

IF 00 <= YY <= 70 THEN

The date is interpreted as 20YY

Issuing Options

The following table identifies the eligible options for issuing the CONNECT command:

Table 15. How the CONNECT Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	Yes	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To list a user's connections groups, see “LISTUSER (List User Profile)” on page 223.
- To list the users connected to a group, see “LISTGRP (List Group Profile)” on page 214.
- To remove a user from a group, see “REMOVE (Remove User from Group)” on page 376.

Authorization Required

The specified users and group must already be defined to RACF.

When issuing the CONNECT command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

To use the CONNECT command, you must have at least one of the following:

CONNECT

- The SPECIAL attribute
- The group-SPECIAL attribute in the group
- The ownership of the group
- JOIN or CONNECT authority in the group

You cannot give a user a higher level of authority in the group than you have.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

Note: If a user is added to a RACF group as a result of a CONNECT command while the user is logged on, the user must logoff and logon again to use that authority to access resources in classes that have been RACLISTed. In addition, started tasks have to STOP and START to use the new authority. This might include started tasks such as JES2 or JES3.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the CONNECT command is:

```
[subsystem-prefix]{CONNECT | CO}
                        (userid ...)
                        [ ADSP | NOADSP ]
                        [ AT( [ node ] .userid ...) | ONLYAT( [ node ] .userid ...) ]
                        [ AUDITOR | NOAUDITOR ]
                        [ AUTHORITY(group-authority) ]
                        [ GROUP(group-name) ]
                        [ GRPACC | NOGRPACC ]
                        [ OPERATIONS | NOOPERATIONS ]
                        [ OWNER(userid or group-name) ]
                        [ RESUME [ (date)] ]
                        [ REVOKE [ (date)] ]
                        [ SPECIAL | NOSPECIAL ]
                        [ UACC [ (access-authority) ] ]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF

subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

userid

specifies the RACF-defined user to be connected to, or modified in, the group specified in the GROUP operand. If you are specifying more than one user, you must enclose the user IDs in parentheses.

In general, the maximum number of users you can connect to one group is 5957. See *z/OS Security Server RACF Macros and Interfaces* for information about how to determine the exact maximum number.

The exception to this is a group that has been defined as a UNIVERSAL group. A UNIVERSAL group may have an unlimited number of users, with USE authority, connected to it for the purpose of resource access.

The number of users in a universal group with authority higher than USE, or with the attributes SPECIAL, OPERATIONS or AUDITOR at the group level, is still limited to 5957.

When displayed with the LISTGRP command, not all members of a UNIVERSAL group will be listed. Only users with authority higher than USE or with the attributes SPECIAL, OPERATIONS or AUDITOR at the group level will be shown in the member list.

This operand is required and must be the first operand following CONNECT.

ADSP | NOADSP**ADSP**

specifies that when the user is connected to this group, all permanent tape and DASD data sets created by the user is RACF-protected by discrete profiles.

RACF ignores the ADSP attribute at LOGON/job initiation if SETROPTS NOADSP is in effect.

NOADSP

specifies that the user is not to have the ADSP attribute. If you are creating a connection and omit both ADSP and NOADSP, NOADSP is the default. A user attribute of ADSP specified on the ADDUSER or ALTUSER command overrides NOADSP as a connect attribute.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([node].userid ...)

specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT([node].userid ...)

specifies that the command is to be directed only to the node specified by *node* where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

CONNECT

If *node* is not specified, the command is directed only to the local node.

AUDITOR | NOAUDITOR

AUDITOR

specifies that the user is to have the group-AUDITOR attribute when connected to this group.

To enter the AUDITOR operand, you must have either the SPECIAL attribute or the group-SPECIAL attribute in the group to which you are connecting or modifying the user's profile.

NOAUDITOR

specifies that the user is not to have the group-AUDITOR attribute when connected to this group. When you are creating a connection and omit both AUDITOR and NOAUDITOR, NOAUDITOR is the default. If you are modifying an existing connection, you must have either the SPECIAL attribute or the group-SPECIAL attribute in the group in which you are modifying the user's profile.

A user attribute of AUDITOR specified on the ADDUSER or ALTUSER command overrides NOAUDITOR as a connect attribute.

AUTHORITY(*group-authority*)

specifies the level of authority the user is to have in the group. The valid group authority values are USE, CREATE, CONNECT, and JOIN, as described in "Group authorities" on page 13. If you are creating a connection and omit AUTHORITY or enter it without a value, the default is USE.

You cannot give a user a higher level of authority in the group than you have.

GROUP(*group-name*)

specifies a RACF-defined group. If you omit this operand, the user is connected to or modified in your current connect group.

Note: RACF allows you to connect a user to more than 300 groups, which is the same as NGROUPS_MAX variable defined in the POSIX standard, but when z/OS UNIX group information is requested, only up to the first 300 z/OS UNIX groups that have GIDs are associated with the process or user.

The first 300 z/OS UNIX groups that have GIDs to which a user is connected are used by z/OS UNIX. LISTUSER displays the groups in the order that RACF examines them when determining which of the user's groups are z/OS UNIX groups.

In addition, the number of users connected to a group should be within the limits allowed by the NFS Client for remote access to files. See *z/OS UNIX System Services Planning* for information on NGROUPS_MAX.

GRPACC | NOGRPACC

GRPACC

specifies that when the user is connected to this group, any group data sets defined by the user are automatically accessible to other users in the group. The group whose name is used as the high-level qualifier of the data set name (or the qualifier supplied by a command installation exit) has UPDATE access authority to the data set.

NOGRPACC

specifies that the user is not to have the GRPACC attribute. If you are

creating a connection and omit both GRPACC and NOGRPACC, NOGRPACC is the default. A user attribute of GRPACC specified on the ADDUSER or ALTUSER command overrides NOGRPACC as a connect attribute.

OPERATIONS | NOOPERATIONS

OPERATIONS

specifies that the user is to have the group-OPERATIONS attribute when connected to this group. The group-OPERATIONS user has authorization to do maintenance operations on all RACF-protected DASD data sets, tape volumes, and DASD volumes within the scope of the group unless the access list for a resource specifically limits the OPERATIONS user to an access authority that is less than the operation requires.

You establish the lower access authority for the group-OPERATIONS user through the PERMIT command.

To enter the OPERATIONS operand, you must have the SPECIAL attribute or the group-SPECIAL attribute in the group to which you are connecting or modifying the user's profile.

NOOPERATIONS

specifies that the user is not to have the group-OPERATIONS attribute in this group. If you are creating a connection and omit both OPERATIONS and NOOPERATIONS, NOOPERATIONS is the default. If you are modifying an existing connection, you must have the SPECIAL attribute or the group-SPECIAL attribute in the group in which you are modifying the user's profile.

A user attribute of OPERATIONS specified on the ADDUSER or ALTUSER command overrides NOOPERATIONS as a connect attribute.

OWNER(*userid or group-name*)

specifies a RACF-defined user or group to be assigned as the owner of the connect profile. If you are creating a connection and you do not specify an owner, you are defined as the owner of the connect profile.

RESUME[(*date*)]

specifies that the user, when connected to the group specified on the GROUP operand, is to be allowed to access the system again. You normally use RESUME to restore access to the system that has been prevented by a prior REVOKE operand. (RESUME, using the current date, is also the default when you are using the CONNECT command to create an initial connection between a user and this group.)

If you specify a date, RACF does not allow the user to access the system until the date you specify. The date must be a future date; if it is not, you are prompted to provide a future date.

During the time between when you specify the RESUME and the date when the RESUME takes effect, the RESUME is called a pending RESUME.

You specify a date in the form *mm/dd/yy*, and you need not specify leading zeros; specifying 9/1/94 is the same as specifying 09/01/94. The date must be a future date; if it is not, you are prompted to provide a future date. RACF interprets dates as 20yy when yy is less than 71, and 19yy when yy is 71 or higher. So, 09/01/94 would be in the year 1994, and 09/01/14 would be in the year 2014.

CONNECT

If you specify RESUME without a date, the RESUME takes effect immediately. Specifying RESUME without a date overrides any pending RESUME and any pending REVOKE.

When no REVOKE is in effect for the user, RACF ignores the RESUME operand and issues a message.

Notes:

1. If you use the ALTUSER command to issue a REVOKE for a user, you must use the ALTUSER command to issue the corresponding RESUME. Issuing RESUME on the CONNECT command does not restore access revoked on the ALTUSER command.
2. If you specify both REVOKE(*date*) and RESUME(*date*), RACF acts on them in date order. For example, if you specify RESUME(8/19/94) and REVOKE(8/5/94), RACF prevents the user from accessing the system from August 5, 1994, to August 18, 1994. On August 19, the user can again access the system.

If a user is already revoked and you specify RESUME(8/5/94) and REVOKE(8/19/94), RACF allows the user to access the system from August 5, 1994, to August 18, 1994. On August 19, RACF prevents the user from accessing the system.
3. If RACF detects a conflict between REVOKE and RESUME (for example, you specify both without a date), RACF uses REVOKE. If you specify, without a date, only REVOKE or only RESUME, RACF clears both date fields. When RACF revokes a user because of inactivity or invalid password attempts, RACF also clears both date fields.

REVOKE[(*date*)]

specifies that RACF is to prevent the user from accessing the system by attempting to connect to the group specified on the GROUP operand; the user's profile and data sets, however, are not deleted from the RACF database.

If you specify a date, RACF does not prevent the user from accessing the system until the date you specify. The date must be a future date; if it is not, you are prompted to provide a future date.

You specify a date in the form *mm/dd/yy*, and you need not specify leading zeros; specifying 9/1/94 is the same as specifying 09/01/94. The date must be a future date; if it is not, you are prompted to provide a future date. RACF interprets dates as 20yy when yy is less than 71, and 19yy when yy is 71 or higher. So, 09/01/94 would be in the year 1994, and 09/01/14 would be in the year 2014. During any time between when you specify the REVOKE and the date when the REVOKE takes effect, the REVOKE is called a pending revoke.

If you specify REVOKE without a date, the REVOKE takes effect the next time the user tries to log on to the system. Specifying REVOKE without a date overrides any pending REVOKE. (Note that RESUME works the same way; if you specify RESUME without a date, it also overrides any pending REVOKE.)

When a REVOKE is already in effect for the user, RACF ignores the REVOKE operand and issues a message.

Notes:

1. If you specify both REVOKE(*date*) and RESUME(*date*), RACF acts on them in date order. For example, if you specify RESUME(8/19/94) and REVOKE(8/5/94), RACF prevents the user from accessing the system from August 5, 1994, to August 18, 1994. On August 19, the user can again access the system.

If a user is already revoked and you specify RESUME(8/5/94) and REVOKE(8/19/94), RACF allows the user to access the system from August 5, 1994, to August 18, 1994. On August 19, RACF prevents the user from accessing the system.

2. If RACF detects a conflict between REVOKE and RESUME (for example, you specify both without a date), RACF uses REVOKE. If you specify, without a date, only REVOKE or only RESUME, RACF clears both date fields. When RACF revokes a user because of inactivity or invalid password attempts, RACF also clears both date fields.

SPECIAL | NOSPECIAL

SPECIAL

specifies that the user is to have the group-SPECIAL attribute when connected to this group. To enter the SPECIAL operand, you must have the SPECIAL attribute or the group-SPECIAL attribute in the group to which you are connecting or modifying the user's profile.

NOSPECIAL

specifies that the user is not to have the group-SPECIAL attribute. If you are creating a connection and omit both SPECIAL and NOSPECIAL, NOSPECIAL is the default. If you are modifying an existing connection, you must have the SPECIAL attribute or the group-SPECIAL attribute in the group in which you are modifying the user's profile.

A user attribute of SPECIAL specified on the ADDUSER or ALTUSER command overrides NOSPECIAL as a connect attribute.

UACC[(access-authority)]

specifies the default value for the universal access authority for all new resources the user defines while connected to the specified group. The universal access authorities are ALTER, CONTROL, UPDATE, READ, and NONE. (RACF does not accept EXECUTE access authority with the CONNECT command.) If you are creating a connection and omit UACC or enter it without a value, the default is NONE.

This option is group-related. The user can have a different default universal access authority in each of the groups to which the user is connected. When a user (who has the ADSP attribute or specifies the PROTECT parameter on a JCL DD statement) enters the system using the group specified in the GROUP operand as the current connect group, RACF assigns this default universal access authority to any data set or tape volume RACF profiles the user defines.

Examples

Table 16. CONNECT Examples

Example 1	<i>Operation</i> User WJE10 wants to connect users AFG5 and GMD2 to group PAYROLL and to make PAYROLL the owner of the connect profiles.
	<i>Known</i> User WJE10 has JOIN authority to group PAYROLL.
	User WJE10 is currently connected to group PAYROLL.
	Users AFG5 and GMD2 are defined to RACF but not connected to group PAYROLL.
	User WJE10 wants to issue the command as a RACF TSO command.
<i>Command</i>	CONNECT (AFG5 GMD2) OWNER(PAYROLL)
<i>Defaults</i>	GROUP(PAYROLL) AUTHORITY(USE) UACC(NONE) NOADSP NOGRPACC
	RESUME NOOPERATIONS NOSPECIAL NOAUDITOR

CONNECT

Table 16. *CONNECT Examples (continued)*

Example 2

Operation User WRH0 wants to CONNECT user PDJ6 to group RESEARCH with CREATE authority and universal access of UPDATE. User WRH0 wants to direct the command to run under the authority of user EMWIN at node RALNC.

Known User EMWIN at RALNC has CONNECT authority to group RESEARCH.

RESEARCH is not the default group of user EMWIN at RALNC.

User PDJ6 is defined to RACF on node RALNC but is not connected to group RESEARCH.

User WRH0 wants to issue the command as a RACF TSO command.

WRH0 and EMWIN at RALNC have an already established user ID association.

Command CONNECT PDJ6 GROUP(RESEARCH) AUTHORITY(CREATE) UACC(UPDATE)
AT(RALNC.EMWIN)

Defaults NOGRPACC RESUME NOOPERATIONS NOSPECIAL NOAUDITOR NOADSP
OWNER(WRH0)

Example 3

Operation User IRB01 wants to revoke the user ID of an employee, user D5819, who will be on vacation for three weeks, starting on August 5, 1994.

Known User IRB01 is the owner of the profile for user D5819. Today's date is August 3, 1994. User IRB01 wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.

Command @CONNECT D5819 REVOKE(8/5/94) RESUME(8/26/94)

Defaults None

DELDSD (Delete Data Set Profile)

Purpose

Use the DELDSD command to remove RACF protection from tape or DASD data sets that are protected by either discrete or generic profiles.

When RACF-protection is removed from a data set protected by a discrete profile:

- The RACF indicator for the data set is turned off. For a DASD data set, the indicator is in the DSCB for a non-VSAM data set or in the catalog entry for a VSAM data set. For a tape data set, the indicator is in the TVTOC entry for the data set in the corresponding TAPEVOL profile.
- The data set profile is deleted from the RACF database. (Note that the data set itself is not physically deleted or scratched.)

If all the data sets in the TVTOC have expired, then RACF deletes the TAPEVOL profiles and the associated tape DATASET profiles.

To remove RACF protection from a non-VSAM DASD data set that is protected by a discrete profile, the data set must be online and not currently in use. For a VSAM data set that is protected by a discrete profile, the catalog for the data set must be online. The VSAM data set itself must also be online if the VSAM catalog recovery option is being used. If the required data set or catalog is not online, the DELDSD command processor requests that the volume be mounted if you have the TSO MOUNT authority.

Changes made to discrete profiles take effect after the DELDSD command is processed. Changes made to generic profiles do not take effect until one or more of the following steps is taken:

- The user of the data set issues the LISTDSD command:

```
LISTDSD DA(data-set-protected-by-the-profile) GENERIC
```

Note: Use the data set name, not the profile name.

- The security administrator issues the SETROPTS command:

```
SETROPTS GENERIC(DATASET) REFRESH
```

See SETROPTS command for authorization requirements.

- The user of the data set logs off and logs on again.

Note: For more information, refer to *z/OS Security Server RACF Security Administrator's Guide*.

Issuing Options

The following table identifies the eligible options for issuing the DELDSD command:

Table 17. How the DELDSD Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	Yes	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

DELDSD

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To create a data set profile, see “ADDSD (Add Data Set Profile)” on page 33.
- To change a data set profile, see “ALTDSD (Alter Data Set Profile)” on page 90.
- To display a data set profile, see “LISTSD (List Data Set Profile)” on page 200.
- To obtain a list of data set profiles, see “SEARCH (Search RACF Database)” on page 408.

Authorization Required

When issuing the DELDSD command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

To remove RACF protection from a data set or to delete a generic data set profile, you must have sufficient authority over the data set. RACF performs authorization checking in the following sequence until you meet one of these conditions:

- You have the SPECIAL attribute.
- The data set profile is within the scope of a group in which you have the group-SPECIAL attribute.
- The high-level qualifier of the profile name (or the qualifier supplied by a command installation exit) is your user ID.
- You are the owner of the profile.

For discrete profiles only:

- The data set is protected by a discrete profile, and you are on the access list with ALTER authority.
- The data set is protected by a discrete profile, and your group or one of your groups (if checking list of groups is active) is on the access list and has ALTER authority.
- The data set is protected by a discrete profile, and the universal access authority is ALTER.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the DELDSD command is:

<code>[<i>subsystem-prefix</i>]{DELDSD DD}</code>

(profile-name...)

[AT([*node*] .*userid* ...) | ONLYAT([*node*] .*userid* ...)]

[GENERIC | NOSET | SET]

[VOLUME(*volume-serial*)]

Note: If you specify a profile name containing generic characters, RACF ignores the VOLUME, SET and NOSET operands..

Notes:

1. For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.
2. For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

profile-name...

specifies the name of the discrete or generic profile. If you specify more than one profile, the list must be enclosed in parentheses.

This operand is required and must be the first operand following DELDSD.

Note: Because RACF uses the RACF database and not the system catalog, you cannot use alias data set names.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([node].userid ...)

specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT([node].userid ...)

specifies that the command is to be directed only to the node specified by *node* where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed only to the local node.

GENERIC | NOSET | SET

GENERIC

specifies that RACF is to treat the profile name as a generic name, even if it does not contain any generic characters.

NOSET | SET

specifies whether the RACF indicator should be set off or not.

If the profile name contains a generic character or if you specify GENERIC, RACF ignores this operand.

NOSET

specifies that RACF is not to turn off the RACF indicator for the data set.

Use NOSET when you are transferring a RACF-indicated data set to another system where it is also to be RACF-protected. Leaving the indicator on prevents unauthorized access to the data set until it can be redefined on the new system. (To delete multiple data set profiles, see Example 2 for the SEARCH command.)

When you specify NOSET for a tape data set protected by a discrete profile, RACF deletes the discrete profile but retains the TVTOC entry for the data set name. You can then use a generic profile to protect the data set.

If you specify NOSET, the volumes on which the data set or catalog resides need not be online.

To use NOSET, you must have the SPECIAL attribute, the data set profile must be within the scope of a group in which you have the group-SPECIAL attribute, or the high-level qualifier of the data set name (or the qualifier supplied by the naming conventions table or by a command installation exit) must be your user ID.

SET

specifies that RACF is to turn off the RACF indicator for the data set. Use SET, which is the default value, when you are removing RACF protection for a data set. If the indicator is already off, the command fails.

VOLUME(*volume-serial*)

specifies the volume on which the tape data set, the non-VSAM DASD data set, or the catalog for the VSAM data set resides.

If you specify this operand and *volume-serial* does not appear in the profile for the data set, the command fails.

If the data set name appears more than once in the RACF database and you do not specify VOLUME, the command fails. If the data set name appears only once and you do not specify VOLUME, no volume serial number checking is performed, and processing continues.

If the profile name contains a generic character or if you specify GENERIC, RACF ignores this operand.

Examples

Table 18. DELDSD Examples

Example 1	<i>Operation</i>	User EH0 wants to remove discrete profile RACF protection from data set CD0.DEPT1.DATA. User EH0 wants to direct the command to run at node CPPD0 under the authority of user GCP02 and prohibit the command from being automatically directed to other nodes.
	<i>Known</i>	User GCP02 at CPPD0 owns data set CD0.DEPT1.DATA. User EH0 wants to issue the command as a RACF TSO command. Users EH0 and GCP02 at CPPD0 have an already established user ID association. Users EH0 and GCP02 at CPPD0 have the SPECIAL attribute.
	<i>Command</i>	DELDSD 'CD0.DEPT1.DATA' ONLYAT(CPPD0.GCP02)
	<i>Results</i>	The command is only processed at node CPPD0 and not automatically directed to any other nodes in the RRSF configuration.
Example 2	<i>Operation</i>	User KLE05 wants to enter a RACF TSO command to remove discrete profile protection from data set KLE05.DUPDS1.DATA. The data set is a duplicate data set, and the user wants to remove the profile for the data set on volume DU2 without turning off the RACF indicator.
	<i>Command</i>	DELDSD DUPDS1.DATA VOLUME(DU2) NOSET
	<i>Defaults</i>	None
Example 3	<i>Operation</i>	User JTB01 wants to delete the generic profile and remove RACF protection from the data set or sets protected by the profile SALES.*.DATA
	<i>Known</i>	User JTB01 has the group-SPECIAL attribute in group SALES. User JTB01 wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.
	<i>Command</i>	@DELDSD 'SALES.*.DATA'
	<i>Defaults</i>	None

DELGROUP (Delete Group Profile)

Purpose

Use the DELGROUP command to delete a group and its relationship to its superior group from RACF.

There are, however, other places in the RACF database where the group name might appear, and DELGROUP processing does not delete these other occurrences of the group name. For example, the group name could be in the access list for any resource. You can use the RACF Remove ID utility (IRRRID00) to remove all occurrences of a group name.

The DELGROUP command does not work for a UNIVERSAL group, in most cases. To delete a UNIVERSAL group, the RACF Remove ID Utility (IRRRID00) should be used.

For information on using the RACF Remove ID utility, see *z/OS Security Server RACF Security Administrator's Guide*.

Issuing Options

The following table identifies the eligible options for issuing the DELGROUP command:

Table 19. How the DELGROUP Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	Yes	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To add a group profile to the RACF database, see “ADDGROUP (Add Group Profile)” on page 24.
- To change a group profile in the RACF database, see “ALTGROUP (Alter Group Profile)” on page 104.
- To connect a user to a group, see “CONNECT (Connect User to Group)” on page 173.
- To list information related to a group profile, see “LISTGRP (List Group Profile)” on page 214.
- To remove a user from a group profile, see 376.
- To obtain a list of group profiles, see “SEARCH (Search RACF Database)” on page 408.

Authorization Required

When issuing the DELGROUP command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

To use the DELGROUP command, at least one of the following must be true:

- You must have the SPECIAL attribute
- The group to be deleted must be within the scope of a group in which you have the group-SPECIAL attribute
- You must be the owner of the superior group
- You must have JOIN authority in the superior group
- You must be the owner of the group to be deleted

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the DELGROUP command is:

```
[subsystem-prefix]{DELGROUP | DG}
                        (group-name ...)
                        [ AT( [ node ] .userid ...) | ONLYAT( [ node ] .userid ...) ]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

DELGROUP

group-name

specifies the name of the group whose profile is to be removed from the RACF database. If you are deleting more than one group, you must enclose the list of group names in parentheses.

You must enter at least one group name. For each group name you enter, the following conditions must exist:

- The group must be defined to RACF.
- The group must not have any subgroups.
- The group must not have any group data sets (data sets whose names are qualified by the group name or begin with the value supplied by an installation exit).
- The group must not have any users connected to it.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([*node*].*userid* ...)

specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT([*node*].*userid* ...)

specifies that the command is to be directed only to the node specified by *node* where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed only to the local node.

Examples

Table 20. DELGROUP Example

Example	<p><i>Operation</i> User WJE10 wants to delete subgroups DEPT1 and DEPT2 from group PAYROLL.</p> <p><i>Known</i> User WJE10 has JOIN authority to group PAYROLL.</p> <p>DEPT1 and DEPT2 are subgroups of group PAYROLL.</p> <p>Neither DEPT1 nor DEPT2 have any subgroups or users connected to them. In addition, neither group has any group data sets.</p> <p>User WJE10 wants to issue the command as a RACF TSO command.</p> <p><i>Command</i> DELGROUP (DEPT1 DEPT2)</p> <p><i>Defaults</i> None</p>
----------------	--

DELUSER (Delete User Profile)

Purpose

Use the DELUSER command to delete a user from RACF.

This command removes the user's profile and all user-to-group connections for the user. (The connect profiles define the user's connections to various RACF groups.)

There are, however, other places in the RACF database where the user's user ID might appear, and the DELUSER command does not delete the user ID from all these places. Specifically, the user could be the owner of a group, the owner of a user's profile, the owner of a group data set, or in an access list for any resource. Before issuing DELUSER, you must first issue the REMOVE command to assign new owners for any group data sets the user owns in groups other than his default group. You can use the RACF Remove ID utility (IRRRID00) to remove all of the occurrences of a user ID. For information on using the RACF Remove ID utility, see *z/OS Security Server RACF Security Administrator's Guide*.

You can use the DELUSER command to delete a TSO user from the RACF database. However, you have no way of knowing if the TSO user is logged on to TSO at the time you issue the DELUSER command. As a result, if the user is logged on to TSO, the user remains active until logging off. Therefore, you might consider having the console operator examine any logons (or jobs) that are active for the TSO user and cancel those that should not be allowed to continue.

The DELUSER command supports digital certificates. If the command issuer is authorized to delete the user profile, and the DELUSER command processor has decided that the user profile can be deleted, the profiles in the DIGTCERT, DIGTRING, or DIGTNMAP classes that describe certificates, private key information, key rings, or certificate mappings associated with the user profile are also deleted. When determining what certificates to delete, the list of certificates from the user profile is used. Certificates that are to be deleted as a result of DELUSER processing are removed from any rings they are connected to at the time the DELUSER command was issued. Likewise, rings that are to be deleted as a result of DELUSER processing have all certificates connected to them removed prior to being deleted. No additional authority checking is done. Authority to the IRR.DIGTCERT.*function* resource is not required. If an error is encountered by DELUSER while attempting to delete a DIGTCERT, DIGTRING, or DIGTNMAP profile, the DELUSER command is terminated without attempting to delete the user profile. If the error indicates that the template is down-level, an error message is issued and the user profile is deleted.

Note: User IDs with mixed-case characters, such as `irrcerta`, `irrsitec`, and `irmulti` which are associated with digital certificates, cannot be specified as *userid* in the DELUSER command because DELUSER cannot process mixed-case user IDs.

Issuing Options

The following table identifies the eligible options for issuing the DELUSER command:

DELUSER

Table 21. How the DELUSER Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	Yes	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To add a user profile to the RACF database, see “ADDUSER (Add User Profile)” on page 48.
- To change a user profile in the RACF database, see “ALTUSER (Alter User Profile)” on page 115.
- To list information in a user profile, see “LISTUSER (List User Profile)” on page 223.
- To administer user ID associations, see “RACLINK (Administer User ID Associations)” on page 296.

Authorization Required

When issuing the DELUSER command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

To use the DELUSER command, at least one of the following must be true:

- You must have the SPECIAL attribute.
- The user profile to be deleted must be within the scope of a group in which you have the group-SPECIAL attribute.
- You must be the owner of the user's profile.

Note: JOIN authority in the user's default group is not sufficient authority to delete the user from RACF.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the DELUSER command is:

```
[subsystem-prefix]{DELUSER | DU}
                (userid ... )
                [ AT( [ node ] .userid ...) | ONLYAT( [ node ] .userid ...) ]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

userid

specifies the user ID of the user whose profile is to be deleted from the RACF database. If you are deleting more than one user, you must enclose the list of user IDs in parentheses. You must enter at least one user ID. For each user ID you enter, the following conditions must exist:

- The user must be defined to RACF.
- The user must not have any user data sets defined to RACF. (User data sets are data sets whose names are qualified by the user ID of the user being deleted or begin with the value supplied by an installation exit.)
- The user cannot have any user ID associations defined. User ID associations for a user must be deleted before the user can be deleted.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([*node*].*userid ...*)

specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT([*node*].*userid ...*)

specifies that the command is to be directed only to the node specified by *node* where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed only to the local node.

DELUSER

Examples

Table 22. DELUSER Examples

Example 1

Operation User WJE10 wants to delete user AEH0 from RACF.

Known User AEH0 is defined to RACF.

User AEH0 is not the owner of any RACF profiles.

User WJE10 is connected to group PAYROLL (and is the owner of user AEH0) with the group-SPECIAL attribute.

User WJE10 wants to issue the command as a RACF TSO command.

Command DELUSER AEH0

Defaults None

Example 2

Operation User SPB1 wants to delete user CA00 from RACF.

Known User CA00 is defined to RACF.

User SPB1 is not the owner of any RACF profiles.

User SPB1 is connected to group PAYROLL (and is the owner of user CA00) with the group-SPECIAL attribute.

User SPB1 wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.

Command @DELUSER CA00

Defaults None

DISPLAY (Display Signed-On-From List)

Background

Persistent verification allows users to sign on to a partner LU (logical unit) and have their authority persist. In other words, once a user has signed on, a password is not required for subsequent sign-on attempts.

APPC/MVS invokes RACF to create and maintain a list called the signed-on-from list. If persistent verification is being used, the signed-on-from list consists of the users currently signed on with persistent verification authority.

Purpose

The RACF DISPLAY operator command displays information held in the signed-on-from list. Entries in the signed-on-from list possess the following information:

- User ID
- Group
- APPL (the local LU name)
- POE (the partner LU name from which the user is signed on)
- SECLABEL

The DISPLAY command has operands which correspond to the items listed above. You can use these operands to select which user entries to display from the signed-on-from list.

The information is displayed as a list of entries sorted by local LU. If there are multiple entries for a given local LU, these entries are sorted by user ID.

Issuing Options

The following table identifies the eligible options for issuing the DISPLAY command:

Table 23. How the DISPLAY Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
No	Yes	No	No	Yes

Note: For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

Use the SIGNOFF command to remove users from the signed-on-from list.

Authorization Required

When issuing the DISPLAY command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

DISPLAY

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The syntax of the DISPLAY command is:

<i>subsystem-prefix</i> DISPLAY	[<u>SIGNON</u>]
	[APPL(<i>local-luname</i> *)]
	[POE(<i>partner-luname</i> *)]
	[USER(<i>userid-name</i> *)]
	[GROUP(<i>group-name</i> *)]
	[SECLABEL(<i>security-label</i> *)]

Note: For additional information on issuing this command as a RACF operator command, refer to “Rules for entering RACF operator commands” on page 21.

Parameters

subsystem-prefix

The subsystem prefix identifies that the RACF subsystem is the processing environment. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

SIGNON

This operand indicates that the information to be displayed is from the signed-on-from list. Since this is always the case, this operand is a default value and can be omitted from the command line.

The operands below allow the operator to select the necessary search criteria. These operands are all optional.

- If none of the operands are specified, you receive an informational message, indicating the version, release and modification level for RACF.
- If no local LU is currently active, you receive an informational message.
- If you specify the APPL operand and at least one local LU is currently active, you receive an informational message with the names of the LU applications listed.

APPL(*local-luname* | *)

The *local-luname* is a 1-8 character name of the local LU to be searched for. An asterisk can occupy the last position of the name in order to provide a partial generic selection capability. A character string consisting of a single asterisk is permitted as a full generic that matches any APPL name in the signed-on-from list. A single asterisk is the default value.

POE(*partner-luname* | *)

The *partner-luname* is the name of the partner LU to be searched for. It can be a 1-8 character unqualified LU name or a 1-17 character network qualified LU name in the format *netid.luname*, where *netid* and *luname* are each 1-8 characters. If the *netid* is omitted, all partner LUs with the specified LU name

portion is shown (POE(LU1) would show NET1.LU1 and NET2.LU1). An asterisk can occupy the last position of the *partner-luname* in order to provide a partial generic selection capability. For example, the *partner-luname* NW1.LU2 would match with *, N*, NW*, NW1*, NW1.*, NW1.L*, NW1.LU*, NW1.LU2*, L*, LU*, and LU2*. A character string consisting of a single asterisk is permitted as a full generic that matches any POE name in the *signed-on-from list*. A single asterisk is the default if another operand (besides SIGNON) is specified.

USER(*userid-name* | *)

The *userid-name* is a 1-8 character name that represents the RACF user ID to be searched for. An asterisk can occupy the last position of the *userid-name* in order to provide a partial generic selection capability. A character string consisting of a single asterisk is permitted as a full generic that matches any user ID in the signed-on-from list. A single asterisk is the default value if either the GROUP operand or the SECLABEL operand is specified.

GROUP(*group-name* | *)

The *group-name* is a 1-8 character name of the RACF group to be searched for. An asterisk can be in the last position of the *group-name* in order to provide a partial generic selection capability. A character string consisting of a single asterisk is permitted as a full generic that matches any group name in the signed-on-from list. A single asterisk is the default value if either the USER operand or the SECLABEL operand is specified. Note that entries in the signed-on-from list might not always be added to that list with a *group-name* value. Such entries have *group-names* consisting of blanks.

SECLABEL(*security-label* | *)

The *security-label* is a 1-8 character name which represents the RACF security label to be searched for. An asterisk can occupy the last position of the specification in order to provide a partial generic selection capability. A character string consisting of a single asterisk is permitted as a full generic that matches any seclabel in the signed-on-from list. A single asterisk is the default value if either the USER operand or the GROUP operand is specified.

Examples

Table 24. DISPLAY Examples

Example 1	<p><i>Operation</i> Display all the partner LUs associated with a particular local LU.</p> <p><i>Known</i> The local LU name is locallu. The RACF subsystem prefix is @.</p> <p><i>Command</i> @display appl(locallu),poe(*)</p> <p><i>Defaults</i> SIGNON</p> <p><i>Output</i> See Figure 3.</p>
Example 2	<p><i>Operation</i> Display all the users signed on for a particular LU pair.</p> <p><i>Known</i> The local LU is locallu, the partner LU is prtnrlu1. The RACF subsystem prefix is @.</p> <p><i>Command</i> @display appl(locallu),poe(prtnrlu1),user(*)</p> <p><i>Defaults</i> SIGNON, GROUP(*), and SECLABEL(*)</p> <p><i>Output</i> See Figure 4.</p>
Example 3	<p><i>Operation</i> Display each local LU and its associated partner LUs, and for each LU pair, display the users signed on.</p> <p><i>Known</i> The RACF subsystem prefix is @.</p> <p><i>Command</i> @display appl(*),poe(*),user(*)</p> <p><i>Defaults</i> SIGNON, GROUP(*), and SECLABEL(*)</p> <p><i>Output</i> See Figure 5.</p>

Attention: In many instances, this command might generate large amounts of display output.

DISPLAY

Table 24. DISPLAY Examples (continued)

- Example 4** *Operation* Display each local LU and its associated partner LUs, and for each LU pair, display the users with *userid_names* beginning with “B”
 Known The RACF subsystem prefix is @.
 Command @display appl(*),poe(*),user(B*),group(*)
 Defaults SIGNON and SECLABEL(*)
 Output See Figure 6.
- Example 5** *Operation* Display all the LU pairs that users have signed on to using a particular group.
 Known The RACF subsystem prefix is @. The *group-name* is grp1.
 Command @display group(grp1),appl(*),poe(*),user(*)
 Defaults SIGNON, SECLABEL(*)
 Output See Figure 7.

```

IRRD004I RACF 2.6.0 SUBSYSTEM 219
REMOTE LU NAME(S) ASSOCIATED WITH ACTIVE LOCAL LU NAME LOCALLU
LU NAME                LU NAME                LU NAME
PRTNRLU1                PRTNRLU2                PRTNRLU3
NETID1.PRTNRLU4

```

Figure 3. Example 1: Output for the DISPLAY Command

```

IRRD004I RACF 2.6.0 SUBSYSTEM 239
LOCAL LU LOCALLU FOR REMOTE LU PRTNRLU1 HAS USER(S):
USER = BOB      GROUP = SYS1    SECLABEL =
USER = BRIAN    GROUP = SYS1    SECLABEL =
USER = JIM      GROUP = GRP1    SECLABEL =
USER = JOE      GROUP = GRP1    SECLABEL =

```

Figure 4. Example 2: Output for the DISPLAY Command

```

IRRD004I RACF 2.6.0 SUBSYSTEM 245
LOCAL LU LOCALLU FOR REMOTE LU PRTNRLU1 HAS USER(S):
USER = BOB      GROUP = SYS1    SECLABEL =
USER = BRIAN    GROUP = SYS1    SECLABEL =
USER = JIM      GROUP = GRP1    SECLABEL =
USER = JOE      GROUP = GRP1    SECLABEL =
LOCAL LU LOCALLU FOR REMOTE LU PRTNRLU2 HAS USER(S):
USER = BRIAN    GROUP =          SECLABEL =
LOCAL LU LOCALLU FOR REMOTE LU PRTNRLU3 HAS USER(S):
USER = BRIAN    GROUP =          SECLABEL =
LOCAL LU LOCALLU FOR REMOTE LU PRTNRLU4 HAS USER(S):
USER = BRIAN    GROUP =          SECLABEL =
LOCAL LU LOCLLU2 FOR REMOTE LU PRTNRLU1 HAS USER(S):
USER = JIM      GROUP = GRP1    SECLABEL =
LOCAL LU LOCLLU3 FOR REMOTE LU PRTNRLU1 HAS USER(S):
USER = JIM      GROUP = GRP1    SECLABEL =

```

Figure 5. Example 3: Output for the DISPLAY Command

```

IRRD004I RACF 2.6.0 SUBSYSTEM 647
LOCAL LU LOCALLU FOR REMOTE LU PRTNRLU1 HAS USER(S):
USER = BOB      GROUP = SYS1  SECLABEL =
USER = BRIAN    GROUP = SYS1  SECLABEL =
LOCAL LU LOCALLU FOR REMOTE LU PRTNRLU2 HAS USER(S):
USER = BRIAN    GROUP =      SECLABEL =
LOCAL LU LOCALLU FOR REMOTE LU PRTNRLU3 HAS USER(S):
USER = BRIAN    GROUP =      SECLABEL =
LOCAL LU LOCALLU FOR REMOTE LU PRTNRLU4 HAS USER(S):
USER = BRIAN    GROUP =      SECLABEL =

```

Figure 6. Example 4: Output for the DISPLAY Command

```

IRRD004I RACF 2.6.0 SUBSYSTEM 251
LOCAL LU LOCALLU FOR REMOTE LU PRTNRLU1 HAS USER(S):
USER = JIM      GROUP = GRP1  SECLABEL =
USER = JOE      GROUP = GRP1  SECLABEL =
LOCAL LU LOCLLU2 FOR REMOTE LU PRTNRLU1 HAS USER(S):
USER = JIM      GROUP = GRP1  SECLABEL =
LOCAL LU LOCLLU3 FOR REMOTE LU PRTNRLU1 HAS USER(S):
USER = JIM      GROUP = GRP1  SECLABEL =

```

Figure 7. Example 5: Output for the DISPLAY Command

HELP (Obtain RACF Help)

Purpose

Use the HELP command to obtain information about the function, syntax, and operands of RACF TSO commands. This information is displayed at your terminal in response to your request for help.

Note: When you use the HELP command to display the syntax for a RACF command, the brackets and braces shown in the syntax diagrams in this book are not displayed on your terminal, and a blank can appear in place of a bracket or brace. If you are unsure whether an operand is optional or required, you should refer to the syntax diagrams contained in this book.

Authorization Required

You need no special attribute or authority to use the HELP command. Any user who can log on to TSO can issue this command.

Issuing Options

The following table identifies the eligible options for issuing the HELP command:

Table 25. How the HELP Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	No	No	No	No

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the HELP command is:

{ HELP H }	[command-name]
	[<u>ALL</u>]
	[FUNCTION]
	[OPERANDS [(operand...)]]
	[SYNTAX]

Parameters

command-name

specifies the name of the command about which you want information.

If you specify this operand, it must be the first operand following HELP. If you omit this operand, you obtain a list of all the TSO commands. However, this list does not contain any RACF commands. Therefore, if you want information about a particular RACF command, you must specify the RACF command name following HELP. Help information is available for all RACF commands that are documented in this book.

ALL

specifies that you want to see all the available information about the command. This information includes the function, syntax, and operands of the command. If no other operand is specified, ALL is the default value.

FUNCTION

specifies that you want to see information about the purpose and operation of the command.

OPERANDS[(operand...)]

specifies that you want to see information about the operands of the command. When you specify OPERANDS and omit any values, all operands for the command is described. To obtain information about a particular operand, specify that operand within parentheses following OPERANDS. If you specify more than one operand, separate the operand names by either commas or blanks.

SYNTAX

specifies that you want to see information about the proper syntax of the command.

Examples

Table 26. HELP Examples

Example 1	<p><i>Operation</i> User LQJ0 wants to see all available information for the ADDUSER command.</p> <p><i>Known</i> User LQJ0 is RACF-defined.</p> <p><i>Command</i> HELP ADDUSER</p> <p><i>Defaults</i> ALL</p>
Example 2	<p><i>Operation</i> User JXN01 wants to see a description of the AUDIT, ADDMEM, DATA, and SECLEVEL operands for the RDEFINE command.</p> <p><i>Known</i> User JXN01 is RACF-defined.</p> <p><i>Command</i> HELP RDEFINE OPERANDS(AUDIT ADDMEM DATA SECLEVEL)</p> <p><i>Defaults</i> None</p>
Example 3	<p><i>Operation</i> User MJW02 wants to see a description of the function and syntax of the SETROPTS command.</p> <p><i>Known</i> User MJW02 is RACF-defined.</p> <p><i>Command</i> HELP SETROPTS FUNCTION SYNTAX</p> <p><i>Defaults</i> None</p>

LISTDSD (List Data Set Profile)

Purpose

Use the LISTDSD command to list information included in tape and DASD data set profiles. A data set profile consists of a RACF segment and, optionally, a DFP or TME segment. The LISTDSD command provides you with the option of listing information contained in the entire data set profile (all segments), or listing the information contained only in a specific segment of the profile.

You can request the details for any number of profiles by giving the full name of each profile. You can also request the details for all profiles whose names are qualified by specific user IDs, group names, or character strings.

You can use the LISTDSD command to cause the changes to go into effect for the generic profiles after issuing the ADDSD, ALTDSD, or DELDSD commands. LISTDSD places a new copy of the profile in the user's address space.

Details RACF lists from the RACF segment of each profile:

- The level
- The owner
- The type of access attempts (as specified by the AUDIT operand on the ADDSD or ALTDSD command) that are being logged on the SMF data set
- The universal access authority
- Your highest level of access authority
- The group under which the profile was created
- The data set type (tape, VSAM, non-VSAM, or MODEL)
- The retention period for a tape data set
- The type of access attempts (as specified by the GLOBALAUDIT operand on the ALTDSD command) that are being logged on the SMF data set (for auditors only)
- The volume serial number (volser) of the volume on which the data set resides.

For both a single volume and multivolume VSAM data set, the volser represents the volume containing the catalog entry for the data set.

For a non-VSAM data set, the volser represents the volume containing the data set itself. If it is a multivolume non-VSAM data set, a list of volsers is given. The list represents the volumes on which the protected data set resides. They are listed in the order in which they were defined.

- Unit information for the data set (if unit information had been specified in the UNIT operand on the ADDSD or ALTDSD command)
- Installation-defined data as specified on the DATA operand of the ADDSD or ALTDSD command

Note: If your installation is running with maximum security (that is, with SETROPTS MLSTABLE, MLS, MLACTIVE, and SECLABELCONTROL all active and the SECLABEL class active), this information is listed only for those with SPECIAL. If you are not SPECIAL, the following text appears in your output in the installation data field:

* SUPPRESSED *

Additional details listed: You can request the following additional details by using the appropriate LISTDSD operands:

- Historical data, such as the date the data set was:
 - Defined to RACF
 - Last referenced
 - Last updated

(see the HISTORY operand)

- The number of times the data set was accessed by all users for each of the following access authorities:
 - ALTER, CONTROL, UPDATE, READ, EXECUTE.

(see the STATISTICS operand)

Note: These details are not meaningful if resource statistics gathering is bypassed at your installation. For a generic profile, RACF replaces any statistics line with NOT APPLICABLE FOR GENERIC PROFILE.

- The standard access list, which displays:
 - All users and groups authorized to access the data set
 - The level of authority for each user and group
 - The number of times each user has accessed the data set

(see the AUTHUSER operand)

- The conditional access list, which displays the same fields as the standard access list as well as the following fields:
 - The class of the resource
 - The entity name of the resource

(see the AUTHUSER operand)

- The information listed below:
 - The user categories authorized to access the data set
 - The security level required to access the data set
 - The security label required to access the data set

(see the AUTHUSER operand)

- The details RACF lists from the DFP segment of the profile:
 - The user ID or group name of the data set resource owner
- The details RACF lists from the TME segment of the profile:
 - The roles and associated access levels

Note: RACF interprets dates with 2 digit years in the following way. YY represents the 2 digit year.

IF 70 < YY <= 99 THEN

The date is interpreted as 19YY

IF 00 <= YY <= 70 THEN

The date is interpreted as 20YY

LISTDSD

Issuing Options

The following table identifies the eligible options for issuing the LISTDSD command:

Table 27. How the LISTDSD Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	No	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To list a general resource profile, see “RLIST (List General Resource Profile)” on page 382. (General resources include terminals and other resources defined in the class descriptor table.)
- To list a user profile, see “LISTUSER (List User Profile)” on page 223.
- To list a group profile, see “LISTGRP (List Group Profile)” on page 214.
- To obtain a list of data set profiles, see “SEARCH (Search RACF Database)” on page 408.

Authorization Required

When issuing the LISTDSD command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

Listing the RACF segment of a data set profile:

To list the details of the RACF segment of a data set profile, you must have a sufficient level of authority for each profile to be listed. One of the following conditions must be met for each profile to be listed:

- You have the SPECIAL attribute.
- The profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You have the OPERATIONS attribute.
- The profile is within the scope of a group in which you have the group-OPERATIONS attribute.

- The high-level qualifier of the profile name (or the qualifier supplied by a command installation exit) is your user ID.
- You are the owner of the profile.
- You are on the profile's access list with at least READ authority. (If your level of authority is NONE, the data set is not listed.)
- Your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is in the access list and has at least READ authority.
- The universal access authority is at least READ.
- You have at least READ access for the profile name from the GLOBAL ENTRY TABLE (if this table contains an entry for the profile).
- You have the AUDITOR attribute.
- The data set profile is within the scope of a group in which you have the group-AUDITOR attribute.

To display the type of access attempts (as specified by the GLOBALAUDIT operand on the ALTDSD command) that are being logged on the SMF data set, either you must have the AUDITOR attribute or the profile must be within the scope of a group in which you have the group-AUDITOR attribute.

To specify the AUTHUSER operand to display the access list for a profile, one of the following conditions must be met for each profile to be listed:

- You have the SPECIAL attribute.
- You have the OPERATIONS attribute.
- You have the AUDITOR attribute.
- The profile is within the scope of a group in which you have the group-SPECIAL attribute.
- The profile is within the scope of a group in which you have the group-OPERATIONS attribute.
- The data set profile is within the scope of a group in which you have the group-AUDITOR attribute.
- The high-level qualifier of the profile name (or the qualifier supplied by a command installation exit) is your user ID.
- You are the owner of the profile.
- You have ALTER access for the profile name from the GLOBAL ENTRY TABLE (if this table contains an entry for the profile).

For discrete profiles only:

- You are on the profile's access list with ALTER authority. (If you have any other level of authority, you cannot use the operand.)
- Your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is in the access list and has ALTER authority. (If any group that RACF checked has any other level of authority, you can not use the operand.)
- The universal access authority is ALTER.

Listing the DFP or TME segment of a data set profile: To list information within the segment of a data set profile, one of the following conditions must be true:

- You have the SPECIAL or AUDITOR attribute.
- You have at least READ authority to the desired field within the segment through field-level access control.

LISTDSD

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the LISTDSD command is:

```
[subsystem-prefix]{LISTDSD | LD}  
  
[ ALL ]  
[ AT( [ node ] .userid ...) | ONLYAT( [ node ] .userid ...) ]  
[ AUTHUSER ]  
[ {DATASET(profile-name ...) | ID(name ...) | PREFIX(char ...)} ]  
[ DFP ]  
[ DSNS ]  
[ GENERIC | NOGENERIC ]  
[ HISTORY ]  
[ NORACF ]  
[ STATISTICS ]  
[ TME ]  
[ VOLUME(volume-serial ...) ]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

ALL

specifies that you want RACF to display all information for each data set.

The access list is included only if you have sufficient authority to use the AUTHUSER operand (see “Authorization Required” on page 202). The type of access attempts (as specified by the GLOBALAUDIT operand on the ALTDSD command) that are being logged on the SMF data set is included only if you have the AUDITOR or group-AUDITOR attribute.

The DFP and TME segments must be requested explicitly.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([*node*].*userid* ...)

specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT(*[node].userid ...*)

LISTDSD is not eligible for automatic command direction. If you specify the ONLYAT keyword, the effect is the same as if you specified the AT keyword.

AUTHUSER

specifies that you want the following information included in the output:

- The user categories authorized to access the data set
- The security level required to access the data set
- The security label required to access the data set
- The standard access list. This contains the following:
 - All users and groups authorized to access the data set
 - The level of authority for each user and group
 - The number of times each user has accessed the data set. This detail is only meaningful when your installation is gathering resource statistics. This detail is not included in the output for generic profiles.
- The conditional access list. This list consists of the same fields as in the standard access list, as well as the following fields:
 - The class of the resource through which each user and group can access the data set. For example, if a user can access the data set through terminal TERM01, then TERMINAL would be the class listed.
 - The entity name of the resource through which each user and group can access the data set. In the example above, TERM01 would be listed.

You must have sufficient authorization to use the AUTHUSER operand (see “Authorization Required” on page 202).

DATASET | ID | PREFIX

DATASET(*profile-name ...*)

specifies the names of one or more data sets whose profiles RACF is to list. If a specified name appears more than once in the RACF database, LISTDSD displays information about all the profiles with that name to which you have proper authority.

The data set name you specify must be enclosed in single quotes unless it is your own data set.

Because RACF uses the RACF database and not the catalog when searching for data set profiles, you cannot use alias data set names.

Note that if you are using naming convention processing, either through the naming convention table or an exit, the name you type might not be the same as the name that appears in the output.

ID(*name ...*)

specifies one or more user IDs or group names. All users and groups must be defined to RACF. Details are listed for all discrete and generic profiles that have the specified user IDs or group names as the high-level qualifier name (or as the qualifier supplied by a command installation exit).

If you do not specify DATASET, PREFIX, or ID, RACF uses your user ID as the default value for the ID operand.

PREFIX(*char ...*)

specifies one or more character strings. Details are listed for all profiles whose names begin with the specified character strings.

LISTDSD

Note that comparison between the character strings and the profile names is not limited to the high-level qualifier. For example, if you specify PREFIX(A.B.C), RACF would display information for profiles such as A.B.C, A.B.CAD, and A.B.C.X.

DFP

specifies that for a DFP-managed data set, you want to list the user ID or group name designated as the data set resource owner. (The data set resource owner, or RESOWNER, is distinguished from the OWNER, which represents the user or group that owns the data set profile.)

DSNS

specifies that you want to list the cataloged data sets protected by the profile specified by the DATASET, ID, or PREFIX operand.

Only data sets cataloged in an Integrated Catalog Facility (ICF) catalog are listed.

Affected tape data sets are listed, regardless of what is specified for SETROPTS TAPEDSN, or whether the TAPEVOL class is active.

When data and index components of VSAM clusters are listed, they are followed by (D) or (I), respectively.

This operand might give unpredictable results if one of the following is true:

- You are using naming convention processing, either through the naming convention table or an exit, to modify data set names so they are protected by different profiles.
- You are using the PREFIX operand of SETROPTS to provide a high-level qualifier for data sets that have only one level in their names.
- There are migrated items in the list and either information about the item cannot be obtained from the migration facility or the migration facility is not available.

In these cases, RACF cannot verify that the item is protected by the input profile and the migrated item is included in the list and is followed by the ? character. Whenever these items are included in the list, the following message appears at the end of the list to explain the ? character.

? = Migrated and unable to verify protection

Notes:

1. If a migrated cluster name appears in the list, but it has an alternate index or path, information on its data or index names is unavailable without recalling the cluster. This message appears after the cluster name:

```
** Migrated cluster component information  
** not available without recall.
```
2. If a migrated cluster name appears in the list and LISTDSD cannot obtain the index and data names due to a migration facility error, this message appears after the cluster name:

```
** Migrated cluster component information  
** not available.
```
3. If the name of a non-migrated cluster appears in the list and RACF is unable to obtain the data and index names specifically through this item, this message appears after the cluster name:

```
** Cluster component information  
** not available.
```

4. If the LISTDSD processor could not obtain all the information on one of the data sets potentially protected by the input profile, it includes the data set in the command output, but follow it with this message:

**** Data set information not available.(x)**

It is likely that this condition occurred because the data set was deleted between the time the LISTDSD DSNS processor first found the names of all the data sets potentially protected by the input profile and the time it processed that particular data set. If that is the case, ignore that data set entry. If that is not the case, issue the LISTDSD command again and if the additional message still appears, contact IBM support; (x) is a numeric value that denotes diagnostic information used by IBM support.

5. The LISTDSD command processor does not include the following items in the output list of protected data sets:
 - Master catalog
 - Alternate Index (AIX) and its components
 - catalogs

GENERIC | NOGENERIC

GENERIC

specifies that RACF is to list only information for the generic profiles. If you specify GENERIC with DATASET, RACF lists information for generic profiles whose names most closely match the data set names you specify.

GENERIC, when specified with DATASET, causes changes to take effect after adding, changing, or deleting generic profiles. It places a fresh copy of the profile in the command user's address space.

NOGENERIC

specifies that RACF is to list only information for discrete profiles.

Notes:

1. If you specify ID or PREFIX but omit GENERIC and NOGENERIC, RACF lists information for all discrete and generic profiles of the data sets owned or represented by the names specified in the command.

For example, if you enter the following command:

```
LISTDSD ID(SMITH)
```

RACF lists all data set profiles for user ID SMITH.

2. If you specify the DATASET operand but omit GENERIC and NOGENERIC, RACF lists information for the discrete profile, if it exists, and the fully-qualified generic profile if it exists, or the generic profile that is not fully-qualified, if its name, including all its qualifiers, matches the name specified on the command.

For example, if you enter the following command:

```
LISTDSD DATASET('XXX.YYY','AA.*')
```

RACF lists information for the discrete profile XXX.YYY, if it exists, the fully-qualified generic profile XXX.YYY if it exists, and the generic profile AA.* if it exists.

3. If you specify DATASET with a fully-qualified name for a data set that is protected by a generic profile that is not fully-qualified, information for this profile can be listed only when GENERIC is specified.

LISTDSD

If you specified DATASET without GENERIC and NOGENERIC and you received an informational message (No RACF description found) for one of the specified fully-qualified names, you might want to retry the command on this name using GENERIC, because it is possible that this data set is protected by a generic profile that is not fully-qualified.

For example, data set BBB.CCC is protected by a generic profile BBB.*. If you enter the following command:

```
LISTDSD DATASET('BBB.CCC')
```

RACF lists information only if there is a discrete profile BBB.CCC, or a fully-qualified generic profile BBB.CCC, or both. But if you enter the following command:

```
LISTDSD DATASET('BBB.CCC') GENERIC
```

RACF lists information for the fully-qualified generic profile BBB.CCC if it exists, or the generic profile that most closely matches BBB.CCC. In this example, the generic profile BBB.* is listed.

4. If generic profile command processing is inactive, only discrete profiles are listed. RACF does not search for generic profiles.

HISTORY

specifies that you want to list the following data:

- The date each profile was defined to RACF
- The date each data set was last referenced
- The date of the last authorization check for UPDATE authority

NORACF

specifies that you want to suppress the listing of RACF segment information from the specified data set's profile. If you specify NORACF, you must include one or more of the following operands: DSNS, DFP, TME.

If you do not specify NORACF, RACF displays the information in the RACF segment of a data set.

The information displayed as a result of using the NORACF operand is dependent on other operands used in the command. For example, if you use NORACF with DSNS, DFP, or TME also specified, only that information (DSNS, DFP, or TME) is displayed.

STATISTICS

specifies that you want to list the statistics for each profile. The list includes the number of times the profile was accessed by users with READ, UPDATE, CONTROL, and ALTER authorities, as well as a separate total for each authority level. These details are meaningful only when your installation is gathering resource statistics. For generic profiles, RACF replaces any statistics line with NOT APPLICABLE FOR GENERIC PROFILE.

TME

specifies that information for the Tivoli Security Management Application is to be listed.

VOLUME(*volume-serial....*)

limits the profiles listed to those found on the specific volume or list of volumes identified by volume serial number. RACF does not list profiles with the same name found on other volumes. If you do not specify NOGENERIC, RACF lists any generic profiles as well.

Examples

Table 28. LISTDSD Examples

Example 1	<i>Operation</i>	User DAF0 wants to list all information for his own data set profiles.
	<i>Known</i>	User DAF0 is RACF-defined, and does not have the AUDITOR attribute. User DAF0 wants to issue the command as a RACF TSO command.
	<i>Command</i>	LISTDSD ALL
	<i>Defaults</i>	ID(DAF0)
	<i>Output</i>	See Figure 8.
Example 2	<i>Operation</i>	User IA0 wants to list the users authorized to data set SYS1.PLIBASE.
	<i>Known</i>	User IA0 has ALTER authority to SYS1.PLIBASE, and does not have the AUDITOR attribute. User IA0 wants to issue the command as a RACF TSO command.
	<i>Command</i>	LISTDSD DATASET('SYS1.PLIBASE') AUTHUSER
	<i>Defaults</i>	None
	<i>Output</i>	See Figure 9.
Example 3	<i>Operation</i>	User ADM1 wants to list a generic profile SALES.*.ABC.
	<i>Known</i>	User ADM1 is the owner of the generic profile, and generic profile command processing is enabled. User ADM1 has the group-AUDITOR attribute in group SALES. User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	LISTDSD DATASET('SALES.*.ABC')
	<i>Defaults</i>	None
	<i>Output</i>	See Figure 10.
Example 4	<i>Operation</i>	User JADAMS wants to display the discrete profile for the DFP-managed data set RESEARCH.TEST.DATA. JADAMS also wants to display the user or group who is the data set resource owner.
	<i>Known</i>	User JADAMS is the owner of the profile protecting data set RESEARCH.TEST.DATA.
		User JADAMS has field-level access of at least READ for the DFP segment.
		User JADAMS wants to issue the command as a RACF TSO command.
	<i>Command</i>	LISTDSD DATASET('RESEARCH.TEST.DATA') DFP
	<i>Defaults</i>	None
	<i>Output</i>	See Figure 11.

LISTDSD

```

LISTDSD ALL
INFORMATION FOR DATASET DAF0.DS2.DATA
LEVEL  OWNER      UNIVERSAL ACCESS  WARNING  ERASE
-----
00      DAF0              READ          NO       NO
AUDITING
-----
SUCCESS(READ),FAILURES(ALTER)
NOTIFY
-----
NO USER TO BE NOTIFIED
YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----
NONE GIVEN      RESEARCH      NON-VSAM
VOLUMES ON WHICH DATASET RESIDES  UNIT
-----
231406                      SYSDA
NO INSTALLATION DATA
          SECURITY LEVEL
-----
NO SECURITY LEVEL
CATEGORIES
-----
NOCATEGORIES
SECLABEL
-----
NO SECLABEL
CREATION DATE  LAST REFERENCE DATE  LAST CHANGE DATE
(DAY)  (YEAR)      (DAY)  (YEAR)      (DAY)  (YEAR)
-----
145    85          145    85          145    85
ALTER COUNT  CONTROL COUNT  UPDATE COUNT  READ COUNT
-----
00000      00010      00000      00010
ID      ACCESS  ACCESS COUNT
-----
IA0      READ      00010
ADM1      READ      00000
PROJECTA  UPDATE      00008
ID      ACCESS  ACCESS COUNT  CLASS  ENTITY NAME
-----
NO ENTRIES IN CONDITIONAL ACCESS LIST

```

Figure 8. Example 1: Output for the LISTDSD Command (Part 1 of 2)

```

INFORMATION FOR DATASET DAF0.DS3.DATA
LEVEL  OWNER      UNIVERSAL ACCESS  WARNING  ERASE
-----
 00    DAF0              READ          NO      NO
AUDITING
-----
ALL(UPDATE)
NOTIFY
-----
NO USER TO BE NOTIFIED
YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----
NONE GIVEN      RESEARCH      NON-VSAM
VOLUMES ON WHICH DATASET RESIDES  UNIT
-----
231406                      SYSDA
NO INSTALLATION DATA
          SECURITY LEVEL
-----
NO SECURITY LEVEL
CATEGORIES
-----
NOCATEGORIES
SECLABEL
-----
NO SECLABEL
CREATION DATE  LAST REFERENCE DATE  LAST CHANGE DATE
(DAY) (YEAR)      (DAY) (YEAR)      (DAY) (YEAR)
-----
 145    85          145    85          145    85
ALTER COUNT  CONTROL COUNT  UPDATE COUNT  READ COUNT
-----
00000          00000          00000          00010
ID      ACCESS  ACCESS COUNT
-----
NO ENTRIES IN STANDARD ACCESS LIST
ID      ACCESS  ACCESS COUNT  CLASS  ENTITY NAME
-----
NO ENTRIES IN CONDITIONAL ACCESS LIST

```

Figure 8. Example 1: Output for the LISTDSD Command (Part 2 of 2)

LISTDSD

```

LISTDSD DATASET('SYS1.PLIBASE') AUTHUSER
INFORMATION FOR DATASET SYS1.PLIBASE
LEVEL  OWNER    UNIVERSAL ACCESS  WARNING  ERASE
-----
00      IA0              READ          NO       NO
AUDITING
-----
SUCCESS(UPDATE)
NOTIFY
-----
NO USER TO BE NOTIFIED
YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----
      ALTER      SYS1          NON-VSAM
VOLUMES ON WHICH DATASET RESIDES  UNIT
-----
231407                      SYSDA
INSTALLATION DATA
-----
PL/1 LINK LIBRARY
      SECURITY LEVEL
-----
NO SECURITY LEVEL
CATEGORIES
-----
NOCATEGORIES
SECLABEL
-----
NO SECLABEL
ID      ACCESS  ACCESS COUNT
-----
ESH25   UPDATE   00009
PROJECTB READ     00015
IA0     ALTER    00020
ID      ACCESS  ACCESS COUNT  CLASS  ENTITY NAME
-----
NO ENTRIES IN CONDITIONAL ACCESS LIST

```

Figure 9. Example 2: Output for the LISTDSD Command

```

LISTDSD DATASET('SALES.*.ABC')
INFORMATION FOR DATASET SALES.*.ABC (G)
LEVEL  OWNER    UNIVERSAL ACCESS  WARNING  ERASE
-----
00      ADM1              READ          NO       NO
AUDITING
-----
ALL(READ)
NOTIFY
-----
NO USER TO BE NOTIFIED
YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----
NONE GIVEN   RESEARCH          NON-VSAM
GLOBALAUDIT
-----
NONE
NO INSTALLATION DATA

```

Figure 10. Example 3: Output for the LISTDSD Command

```

LISTDSD DATASET('RESEARCH.TEST.DATA') DFP
INFORMATION FOR DATASET RESEARCH.TEST.DATA
LEVEL  OWNER    UNIVERSAL ACCESS  WARNING  ERASE
-----
00     JADAMS          READ          NO      NO
AUDITING
-----
ALL(READ)
NOTIFY
-----
NO USER TO BE NOTIFIED
YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----
NONE GIVEN    RESEARCH        NON-VSAM
GLOBALAUDIT
-----
NONE
NO INSTALLATION DATA
DFP INFORMATION
-----
RESOWNER= KSMITH

```

Figure 11. Example 4: Output for the LISTDSD Command

LISTGRP (List Group Profile)

Purpose

Use the LISTGRP command to list details of specific RACF group profiles. A group profile consists of a RACF segment and, optionally, other segments such as DFP and OMVS. The LISTGRP command provides you with the option of listing the information contained in the entire group profile (all segments), or listing the information contained only in a specific segment of the group profile.

The details RACF lists from the RACF segment of each group profile are:

- The superior group of the group
- The owner of the group
- The terminal option of the group
- Whether or not the group is a universal group
- Any subgroups under the group
- Installation-defined data, as specified by the DATA operand of the ADDGROUP and ALTGROUP command
- The name of the data set model profile

RACF lists the following information from the RACF segment of the group profile for each user connected to the group:

- The user ID

An exception to this is when the group is a UNIVERSAL group. When a UNIVERSAL group displayed with the LISTGRP command, not all members will be listed. Only users with authority higher than USE or with the attributes SPECIAL, OPERATIONS or AUDITOR at the group level will be shown in the member list. To view all members of a UNIVERSAL group, the Database Unload Utility (IRRDBU00) must be used. For more information on using the Database Unload Utility (IRRDBU00), see *z/OS Security Server RACF Security Administrator's Guide*.
- The user's level of authority in the group
- The number of times the user has entered the system using this group as the current connect group
- The user's default universal access authority
- The user's connect attributes (group-related user attributes)
- Any REVOKE or RESUME requests either in effect or pending, with the corresponding dates

The details RACF lists from the DFP segment of the group profile are:

- The group's default data class
- The group's default management class
- The group's default storage class
- The data management data application for the group

The details RACF lists from the TME segment of the group profile are:

- The list of roles that refer to this group

The details RACF lists from the OMVS or OVM segment of the group profile are:

- The group's z/OS UNIX System Services group identifier

Issuing Options

The following table identifies the eligible options for issuing the LISTGRP command:

Table 29. How the LISTGRP Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	No	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To list a user profile, see “LISTUSER (List User Profile)” on page 223.
- To list a data set profile, see “LISTDSD (List Data Set Profile)” on page 200.
- To list a general resource profile, see “RLIST (List General Resource Profile)” on page 382. (General resources include terminals and other resources defined in the class descriptor table.)
- To obtain a list of group profiles, see “SEARCH (Search RACF Database)” on page 408.

Authorization Required

When issuing the LISTGRP command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

Listing the RACF segment of a group profile: To list the details of the RACF segment of a group profile, one of the following conditions must be true:

- You have the SPECIAL attribute
- You have the group-SPECIAL attribute in each group to be listed, or each group to be listed is within the scope of a group in which you have the group-SPECIAL attribute
- You have the AUDITOR attribute
- You have the group-AUDITOR attribute in each group to be listed, or each group to be listed is within the scope of a group in which you have the group-AUDITOR attribute
- You are the owner of the group

LISTGRP

- You have JOIN or CONNECT authority in the group

Listing the other segments of a group profile: To list information from segments other than the RACF segment for a group profile, one of the following conditions must be true:

- You have the SPECIAL or AUDITOR attribute
- You have at least READ authority to the desired field through field-level access control

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the LISTGRP command is:

```
[subsystem-prefix]{LISTGRP | LG}  
[ {(group-name ...) | *} ]  
[ AT( [ node ] .userid ...) | ONLYAT( [ node ] .userid ...) ]  
[ DFP ]  
[ NORACF ]  
[ OMVS ]  
[ OVM ]  
[ TME ]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

***group-name* | ***

group-name

Specifies the name of one or more RACF-defined groups. If you specify more than one group name, you must enclose the names in parentheses.

- * Specifies that you want to list information contained in all RACF-defined group profiles to which you have the required authority.

On a system with many groups defined, the use of * may result in a large amount of output and may not be useful to a user issuing the command. It may be more appropriate for the user to browse the output of IRRDBU00 (database unload) or to write a program to process the IRRDBU00 output and produce a report showing only the subset of information that is of

interest to the user. The processing of output of LISTGRP by programs is not supported nor recommended by IBM. If you want a listing of all the groups for use by a program you should instead have the program process the output from IRRDBU00, RACROUTE REQUEST=EXTRACT, or ICHEINTY.

If you specify a group name or *, it must be the first operand following LISTGRP.

If you specify one or more group names (or *) without specifying an additional operand, RACF lists only the RACF segment information from the specified profiles.

If you enter LISTGRP with no operands, RACF lists only the RACF segment information from your current connect group.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([*node*].*userid* ...)

specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT([*node*].*userid* ...)

LISTGRP is not eligible for automatic command direction. If you specify the ONLYAT keyword, the effect is the same as if you specified the AT keyword.

DFP

specifies that you want to list the information contained in the DFP segment of the group profile.

If you specify DFP you must also specify a group name or *.

NORACF

specifies that you want to suppress the listing of base segment information from the group profile. If you specify NORACF, you must also specify one of the other segment names such as DFP or OMVS.

If you do not specify NORACF, RACF displays the information in the RACF segment of a group profile.

OMVS

specifies that you want to list the information contained in the OMVS segment of the group profile.

If you specify OMVS, you must also specify a group name or (*).

If the group profile contains an OMVS segment but GID was not specified on a ADDGROUP or ALTGROUP command, the listing displays the field name followed by the word "NONE".

OVM

specifies that you want to list the information contained in the OVM segment of the group profile.

If you specify OVM, you must also specify a group name or an (*).

LISTGRP

If the group profile contains an OVM segment but GID was not specified on a ADDGROUP or ALTGROUP command, the listing displays the field name followed by the word "NONE".

TME

specifies that information for the Tivoli Security Management Application is to be listed.

If you specify TME you must also specify a group name or an (*).

Examples

Table 30. LISTGRP Examples

Example 1	<i>Operation</i>	User IA0 wants to display the information contained in the RACF segment of the profile for group RESEARCH.
	<i>Known</i>	User IA0 has CONNECT authority to group RESEARCH. User IA0 wants to issue the command as a RACF TSO command.
	<i>Command</i>	LISTGRP RESEARCH
	<i>Defaults</i>	None
Example 2	<i>Output</i>	See Figure 12.
	<i>Operation</i>	User ADM1 wants to display the information contained in the RACF segment of the profiles for all groups.
	<i>Known</i>	User ADM1 has the SPECIAL and AUDITOR attributes. User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	LISTGRP *
Example 3	<i>Defaults</i>	None
	<i>Output</i>	See Figure 13.
	<i>Operation</i>	User ADM1 wants to display the information contained in the RACF segment and DFP segment of the profile for group DFPADMN.
	<i>Known</i>	User ADM1 has the SPECIAL and AUDITOR attributes.
Example 4		Group DFPADMN is defined to RACF, and DFPADMN's profile contains a DFP segment.
		User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	LISTGRP DFPADMN DFP
	<i>Defaults</i>	None
Example 5	<i>Output</i>	See Figure 14.
	<i>Operation</i>	User ADM1 wants to display the information contained in only the DFP segment of the profile for group DFPADMN.
	<i>Known</i>	User ADM1 has the SPECIAL and AUDITOR attributes.
		Group DFPADMN is defined to RACF, and DFPADMN's profile contains a DFP segment.
Example 6		User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	LISTGRP DFPADMN DFP NORACF
	<i>Defaults</i>	None
	<i>Output</i>	See Figure 15.
Example 7	<i>Operation</i>	User ADM1 requests the listing of the OMVS segment for the group OMVSG1.
	<i>Known</i>	User ADM1 has the SPECIAL attribute. User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	LISTGRP OMVSG1 OMVS NORACF
	<i>Defaults</i>	None
Example 8	<i>Output</i>	See Figure 16.

Table 30. LISTGRP Examples (continued)

Example 6

Operation User NETADM requests the listing of the UNIVERSAL group NETGROUP.

Known NETGROUP is a UNIVERSAL group and only users with authority higher than USE or users with SPECIAL, OPERATIONS and AUDITOR at the GROUP level will be displayed in the member list. User NETADM has the SPECIAL attribute to the group NETGROUP. User NETADM wants to issue the command as a RACF TSO command.

Command LISTGRP NETGROUP

Defaults None

Output See Figure 17.

```

LISTGRP RESEARCH
INFORMATION FOR GROUP RESEARCH
SUPERIOR GROUP=SYS1          OWNER=IBMUSER
NO INSTALLATION DATA
NO MODEL DATA SET
TERMUACC
SUBGROUP(S)= PAYROLLB
USER(S)=      ACCESS=      ACCESS COUNT=      UNIVERSAL ACCESS=
IBMUSER      JOIN          000000          ALTER
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE          RESUME DATE=NONE
DAF0      JOIN          000002          READ
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE          RESUME DATE=NONE
IA0      CONNECT      000004          READ
CONNECT ATTRIBUTES=ADSP SPECIAL OPERATIONS
REVOKE DATE=NONE          RESUME DATE=NONE
ESH25      USE          000000          READ
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE          RESUME DATE=NONE
PROJECTB      USE          000000          READ
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE          RESUME DATE=NONE
RV2      CREATE      000000          READ
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE          RESUME DATE=NONE
RV3      CREATE      000000          READ
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE          RESUME DATE=NONE
ADM1      JOIN          000000          READ
CONNECT ATTRIBUTES=OPERATIONS
REVOKE DATE=NONE          RESUME DATE=NONE
AEH0      USE          000000          READ
CONNECT ATTRIBUTES=REVOKED
REVOKE DATE=NONE          RESUME DATE=NONE

```

Figure 12. Example 1: Output for LISTGRP RESEARCH

LISTGRP

```

LISTGRP *
INFORMATION FOR GROUP PAYROLLB
  SUPERIOR GROUP=RESEARCH      OWNER=IBMUSER
  NO INSTALLATION DATA
  NO MODEL DATA SET
  TERMUACC
  NO SUBGROUPS
  USER(S)=      ACCESS=      ACCESS COUNT=      UNIVERSAL ACCESS=
    IBMUSER      JOIN          000000          ALTER
    CONNECT ATTRIBUTES=NONE
    REVOKE DATE=NONE
  DAF0          CREATE          000000          RESUME DATE=NONE
    CONNECT ATTRIBUTES=NONE          READ
    REVOKE DATE=NONE          RESUME DATE=NONE
  IA0          CREATE          000000          READ
    CONNECT ATTRIBUTES=ADSP SPECIAL OPERATIONS
    REVOKE DATE=NONE          RESUME DATE=NONE
  AEH0          CREATE          000000          READ
    CONNECT ATTRIBUTES=NONE
    REVOKE DATE=NONE          RESUME DATE=NONE
INFORMATION FOR GROUP RESEARCH
  SUPERIOR GROUP=SYS1          OWNER=IBMUSER
  NO INSTALLATION DATA
  NO MODEL DATA SET
  TERMUACC
  SUBGROUP(S)= PAYROLLB
  USER(S)=      ACCESS=      ACCESS COUNT=      UNIVERSAL ACCESS=
    IBMUSER      JOIN          000000          ALTER
    CONNECT ATTRIBUTES=NONE
    REVOKE DATE=NONE          RESUME DATE=NONE
  DAF0          JOIN          000002          READ
    CONNECT ATTRIBUTES=NONE
    REVOKE DATE=NONE          RESUME DATE=NONE
  IA0          CONNECT          000004          READ
    CONNECT ATTRIBUTES=ADSP SPECIAL OPERATIONS
    REVOKE DATE=NONE          RESUME DATE=NONE
  ESH25          USE          000000          READ
    CONNECT ATTRIBUTES=NONE
    REVOKE DATE=NONE          RESUME DATE=NONE
  PROJECTB          USE          000000          READ
    CONNECT ATTRIBUTES=NONE
    REVOKE DATE=NONE          RESUME DATE=NONE
  RV2          CREATE          000002          READ
    CONNECT ATTRIBUTES=NONE
    REVOKE DATE=NONE          RESUME DATE=NONE
  RV3          CREATE          000000          READ
    CONNECT ATTRIBUTES=NONE
    REVOKE DATE=NONE          RESUME DATE=NONE
  ADM1          JOIN          000001          READ
    CONNECT ATTRIBUTES=OPERATIONS
    REVOKE DATE=NONE          RESUME DATE=NONE
  AEH0          USE          000000          READ
    CONNECT ATTRIBUTES=NONE
    REVOKE DATE=NONE          RESUME DATE=NONE

```

Figure 13. Example 2: Output for LISTGRP *

```

LISTGRP DFPADMN DFP
INFORMATION FOR GROUP DFPADMN
  SUPERIOR GROUP=SYSADMN      OWNER=SYSADMN
  NO INSTALLATION DATA
  NO MODEL DATA SET
  TERMUACC
  SUBGROUP(S)= DFPGRP1, DFPGRP2
  USER(S)=      ACCESS=      ACCESS COUNT=      UNIVERSAL ACCESS=
  IBMUSER      JOIN      000000      ALTER
    CONNECT ATTRIBUTES=NONE
    REVOKE DATE=NONE      RESUME DATE=NONE
  DSMITH      JOIN      000002      READ
    CONNECT ATTRIBUTES=NONE
    REVOKE DATE=NONE      RESUME DATE=NONE
  HOTROD      CONNECT      000004      READ
    CONNECT ATTRIBUTES=ADSP SPECIAL OPERATIONS
    REVOKE DATE=NONE      RESUME DATE=NONE
  ESHAW      USE      000000      READ
    CONNECT ATTRIBUTES=NONE
    REVOKE DATE=NONE      RESUME DATE=NONE
  PROJECTB      USE      000000      READ
    CONNECT ATTRIBUTES=NONE
    REVOKE DATE=NONE      RESUME DATE=NONE
  ADM1      JOIN      000000      READ
    CONNECT ATTRIBUTES=OPERATIONS
    REVOKE DATE=NONE      RESUME DATE=NONE
  AEHALL      USE      000000      READ
    CONNECT ATTRIBUTES=REVOKED
    REVOKE DATE=NONE      RESUME DATE=NONE
DFP INFORMATION
  MGMTCLASS= DFP2MGMT
  STORCLASS= DFP2STOR
  DATACLAS= DFP2DATA
  DATAAPPL= DFP2APPL

```

Figure 14. Example 3: Output for LISTGRP DFPADMIN DFP

```

LISTGRP DFPADMN DFP NORACF
INFORMATION FOR GROUP DFPADMN
DFP INFORMATION
  MGMTCLASS= DFP2MGMT
  STORCLASS= DFP2STOR
  DATACLAS= DFP2DATA
  DATAAPPL= DFP2APPL

```

Figure 15. Example 4: Output for LISTGRP DFPADMIN DFP NORACF

```

LISTGRP OMVSG1 OMVS NORACF
INFORMATION FOR GROUP OMVSG1
OMVS INFORMATION
  GID= 0000003243

```

Figure 16. Example 5: Output for LISTGRP OMVSG1 OMVS NORACF

LISTGRP

```
EXAMPLE:
LISTGRP NETGROUP
INFORMATION FOR GROUP NETGROUP
  SUPERIOR GROUP=SYS1      OWNER=IBMUSER
  NO INSTALLATION DATA
  NO MODEL DATA SET
  TERMUACC
  UNIVERSAL
  NO SUBGROUPS
  USER(S)=      ACCESS=      ACCESS COUNT=      UNIVERSAL ACCESS=
  IBMUSER      JOIN      00000000      NONE
  CONNECT ATTRIBUTES= NONE
  REVOKE DATE= NONE      RESUME DATE= NONE
  NETADM      CREATE      00000000      READ
  CONNECT ATTRIBUTES= SPECIAL
  REVOKE DATE= NONE      RESUME DATE= NONE
```

Figure 17. Example 6: Output for LISTGRP NETGROUP

LISTUSER (List User Profile)

Purpose

Use the LISTUSER command to list the details of specific RACF user profiles. A user profile consists of a RACF segment and, optionally, other segments such as TSO and DFP. The LISTUSER command provides you with the option of listing the information contained in the entire user profile (all segments), or listing the information contained only in specific segments of the user profile.

You cannot use the LISTUSER command to list information about user ID associations; you must use the RACLINK command.

The details RACF lists from the RACF segment for each user profile are:

- The user ID
- The user's name or UNKNOWN, if the user's name was not specified on the ADDUSER command
- The owner of the user's profile
- The date the user was defined to RACF
- The default group
- The date the user's password was last updated
- The password change interval (in number of days)
- The user's attributes
- The date and time the user last entered the system
- The classes in which the user is authorized to define profiles
- The installation-defined data

Note: If an MVS installation is configured to be a B1 environment, this information is listed in your output. * SUPPRESSED * appears under the installation data field. Only those with SPECIAL will be allowed to list the field.

- The name of default data set model profile
- Any REVOKES or RESUMES either in effect or pending, with the corresponding dates
- The security label, the security level, and category

In addition, RACF lists the following information from the RACF segment of the user profile for each group to which the user is connected:

- The group name
- The user's authority in the group
- The user ID of the person who connected the user to this group
- The date the user was connected to this group
- The number of times the user has entered the system with this group as the current connect group
- The default universal access authority
- The date and time the user last entered the system using this group as the current connect group
- The connect attributes (group-related user attributes)

Note: RACF interprets dates with 2 digit years in the following way. YY represents the 2 digit year.

```
IF 70 < YY <= 99 THEN
```

The date is interpreted as 19YY

```
IF 00 <= YY <= 70 THEN
```

LISTUSER

The date is interpreted as 20YY

Issuing Options

The following table identifies the eligible options for issuing the LISTUSER command:

Table 31. How the LISTUSER Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	No	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To list a group profile, see “LISTGRP (List Group Profile)” on page 214.
- To list a data set profile, see “LISTDSD (List Data Set Profile)” on page 200.
- To list a general resource profile, see “RLIST (List General Resource Profile)” on page 382. (General resources include terminals, and other resources defined in the class descriptor table.)
- To list information about user ID associations, see “RACLINK (Administer User ID Associations)” on page 296.
- To obtain a list of user profiles, see “SEARCH (Search RACF Database)” on page 408.

Authorization Required

When issuing the LISTUSER command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

Listing the RACF segment of a user profile: You can always list the details of the RACF segment of your own user profile. To list details of the RACF segment of another user's profile, one of the following conditions must be true:

- You are the owner of the user's profile.
- You have the SPECIAL attribute.
- The user's profile is within the scope of a group in which you have the group-SPECIAL attribute.

- You have the AUDITOR attribute.
- The user's profile is within the scope of a group in which you have the group-AUDITOR attribute.
- You have READ access to the IRR.LISTUSER resource in the FACILITY class and the user does not have any of the SPECIAL, AUDITOR or OPERATIONS attributes.

Failed access attempts to IRR.LISTUSER are not logged. Rather, these attempts are logged as LISTUSER command violations. Successful accesses to IRR.LISTUSER are logged at the installation's discretion. For more information on logging, see *z/OS Security Server RACF Auditor's Guide*.

To list details of the RACF segment of all RACF-defined user profiles (by specifying the asterisk (*) operand), one of the following conditions must be true for each listed profile:

- You are the owner of the user's profile. RACF lists the RACF segment for all the user profiles that you own.
- You have the SPECIAL attribute. RACF lists the RACF segment for all user profiles.
- The user's profile is within the scope of a group in which you have the group-SPECIAL attribute. RACF lists the RACF segment for all the user profiles within the scope of your group.
- You have the AUDITOR attribute. RACF lists the RACF segment for all user profiles.
- The user's profile is within the scope of a group in which you have the group-AUDITOR attribute. RACF lists the RACF segment for all the user profiles within the scope of your group.
- You have READ access to the IRR.LISTUSER resource in the FACILITY class and the user does not have any of the SPECIAL, AUDITOR or OPERATIONS attributes.

Failed access attempts to IRR.LISTUSER are not logged. Rather, these attempts are logged as LISTUSER command violations. Successful accesses to IRR.LISTUSER are logged at the installation's discretion. For more information on logging, see *z/OS Security Server RACF Auditor's Guide*.

To list a segment other than the base segment, you must have one of the following:

- SPECIAL authority
- AUDITOR authority
- At least READ authority to the desired field within the segment through field-level access control

For information on field-level access checking, see *z/OS Security Server RACF Security Administrator's Guide*.

If you have the group-SPECIAL, AUDITOR, or group-AUDITOR attribute and your installation has assigned security levels and security categories to user profiles, you must have the following to be able to display the RACF segment of a user's profile:

- A security level equal to, or greater than, that in the user profile you are trying to display
- All security categories contained in the user profile you are trying to display contained in your own user profile.

LISTUSER

If you have the AUDITOR attribute, or the profile is within the scope of a group in which you the group-AUDITOR attribute, RACF also lists the value of the UAUDIT/NOUAUDIT operand.

Listing the other segments of a user profile: To list information from segments other than the RACF segment for a user profile, including your own, one of the following conditions must be true:

- You have the SPECIAL or AUDITOR attribute
- You have at least READ authority to the desired field within the segment through field-level access checking.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the LISTUSER command is:

```
[subsystem-prefix]{LISTUSER | LU}
                        [ (userid ...) | * ]
                        [ AT( [ node ] .userid ...) | ONLYAT( [ node ] .userid ...) ]
                        [ CICS ]
                        [ DCE ]
                        [ DFP ]
                        [ EIM ]
                        [ KERB ]
                        [ LANGUAGE ]
                        [ LNOTES ]
                        [ NDS ]
                        [ NETVIEW ]
                        [ NORACF ]
                        [ OMVS ]
                        [ OPERPARM ]
                        [ OVM ]
                        [ PROXY ]
                        [ TSO ]
                        [ WORKATTR ]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

userid | *

userid

Specifies the user ID of one or more RACF-defined users. If you specify more than one user ID, you must enclose the list of user IDs in parentheses.

- * Specifies that you want to list information contained in all RACF-defined user profiles to which you have the required authority.

On a system with many users defined, the use of * may result in a large amount of output that may not be useful to a user issuing the command. It may be more appropriate for the user to browse the output of IRRDBU00 (database unload) or to write a program to process the IRRDBU00 output and produce a report showing only the subset of information that is of interest to the user. The processing of output of LISTUSER by programs is not supported nor recommended by IBM. If you want a listing of all the groups for use by a program you should instead have the program process the output from IRRDBU00, RACROUTE REQUEST=EXTRACT, or ICHEINTY.

Userid or asterisk(*) must be specified if you specify any other operand in the LISTUSER command, and must be the first operand following LISTUSER.

If you enter LISTUSER and specify one or more user IDs (or an asterisk (*)) without specifying an additional operand, RACF lists only the RACF segment information from the specified profiles.

If you enter only LISTUSER, RACF lists only the RACF segment information from your own user profile.

Note: You cannot use the LISTUSER command for user IDs that have mixed-case characters, such as *irrcerta*, *irrsitec*, and *irmulti* (which are associated with digital certificates).

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT(*[node].userid ...*)

specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT(*[node].userid ...*)

LISTUSER is not eligible for automatic command direction. If you specify the ONLYAT keyword, the effect is the same as if you specified the AT keyword.

CICS

specifies that you want to list the information contained in the CICS segment of the user's profile.

The details RACF lists from the CICS segment of the user's profile are:

- The classes assigned to this operator to which BMS messages are sent

Note: The values of the classes are listed in a three digit format, even though a maximum of two digits are used to define the value.

LISTUSER

- Whether the operator are forced off when an XRFSSOFF takeover occurs
- The operator identification
- The priority of the operator
- The time in hours and minutes that the operator is allowed to be idle before being signed off. For systems that are running levels of CICS prior to CICS 4.1, the time is reported in minutes.

DCE

specifies that you want to list the information contained in the DCE segment of the user's profile.

The details that RACF lists from the DCE segment are:

- The DCE universal unique identifier
- The DCE principal name
- The DCE home cell name
- The DCE home cell universal unique identifier
- The DCE AUTOLOGIN indicator

If there is no DCENAME or HOMECCELL for this segment, the field name is not displayed. However, if UUID or HOMEUUID was not specified when the DCE segment was added to the user profile, the word NONE appears in the listing.

DFP

specifies that you want to list the information contained in the DFP segment of the user's profile.

The details RACF lists from the DFP segment of the user's profile are:

- The user's default data class
- The user's default management class
- The user's default storage class
- The data management data application for the user

EIM

Specifies that the Enterprise Identity Mapping (EIM) segment information should be listed.

KERB

Specifies that you want to list the information contained in the KERB segment of the user's profile.

The details RACF lists from the KERB segment of the user's profile are:

- The encryption value settings, For example, ENCRYPT(DES DES3 DESD).
- The local *kerberos-principal-name* (KERBNAME)
- The *max-ticket-life* associated with this local principal (MAXTKTLFE)
- The current Network Authentication Service key version (KEY VERSION)

LANGUAGE

specifies that you want to list the information contained in the LANGUAGE segment of the user's profile.

The 3 character language code and, if defined, the 24 character language name, is displayed. NOT SPECIFIED indicates that no language has been specified.

If the code is displayed without a name, one of the following is true:

- The MVS message service was not active
- The language was not active.

If the language code equals the language name, one of the following is true:

- There was no language name defined on your system
- The language name was defined to be the same as the language code.

The details RACF lists from the LANGUAGE segment of the user's profile are:

- The user's primary language, if one has been specified
- The user's secondary language, if one has been specified

LNOTES

specifies that you want to list the information for the Lotus Notes for z/OS *short-name*, which is contained in the LNOTES segment of the user's profile.

NDS

specifies that you want to list the information for the Novell Directory Services for OS/390 *user-name*, which is contained in the NDS segment of the user's profile.

NETVIEW

specifies that you want to list the information contained in the NETVIEW segment of the user's profile.

The details RACF lists from the NETVIEW segment of the user's profile are:

- The command or command line to be processed by NetView for this operator
- The default MCS console identifier
- Whether security checking should be done for this NetView operator
- Whether this operator can receive unsolicited messages
- The count of operator class values
- The list of NetView scope classes for which this operator has authority
- The number of domains this NetView operator controls
- The list of identifiers of NetView programs in another NetView domain for which this operator has authority
- Whether this operator has administrator authority to the NetView Graphic Monitor Facility (NGMF)

If there is no information in the fields of the NETVIEW segment, the field name is not displayed (with the exception of SIZE, MAXSIZE, and USERDATA).

NORACF

specifies that you want to suppress the listing of RACF segment information from the user's profile.

If you specify NORACF, you must also specify at least one segment name.

The information displayed as a result of using the NORACF operand is dependent on other operands used in the command. For example, if you use NORACF with TSO or DFP also specified, only that information (TSO or DFP) is displayed. User profiles that do not have at least one of the specified segments appear in the command output.

If you do not specify NORACF, RACF displays the information in the RACF segment of a user profile.

OMVS

specifies that you want to list the information contained in the OMVS segment of the user's profile.

The details RACF lists from the OMVS segment are:

- The user identifier
- The initial directory pathname
- The program pathname
- The CPU time, in seconds, the user's processes can use
- The address space region size, in bytes, the user's processes can use
- The maximum number of active or open files the user can have
- The maximum number of active processes the user can have
- The maximum number of threads the user can have

LISTUSER

- The maximum amount of space, in pages, the user can map in storage

Note: If CPUTIMEMAX, ASSIZEMAX, FILEPROCMAx, PROCUSERMAX, THREADSMAX, or MMAPAREAMAX is not specified, or is removed with the ALTUSER command, the word "NONE" appears in the listing. In such situations, z/OS UNIX uses its system level values for limit values.

If there is no HOME or PROGRAM information, the field name is not displayed. However, the word "NONE" appears in the listing if the UID was not specified, or if the UID was removed using the NOUID operand on the ALTUSER command.

OPERPARM

specifies that you want to list the information contained in the OPERPARM segment of the user's profile.

The details RACF lists from the OPERPARM segment of the user's profile are:

- The alternate console group (ALTGRP)
- The operator authority (AUTH)
- Whether the console receives messages that can be automated in a sysplex environment.
- The system name for commands from this console (CMDSYS)
- Whether, and what kind of, delete operator messages are received (DOM)
- The searching key (KEY)
- The message level information (LEVEL)
- Whether system command responses are logged (LOGCMDRESP)
- The message format (MFORM)
- Whether this console is assigned a migration ID (MIGID)
- Event information (MONITOR)
- The systems this console can receive undirected messages from (MSCOPE)
- Routing code information (ROUTCODE)
- Storage information (STORAGE)
- Whether this console receives undeliverable messages (UD)

If there is no information in a field in the user's profile for this segment, the field name is not displayed. However, if no value was specified for STORAGE when the OPERPARM segment was added to the user profile, STORAGE=0 appears in the listing.

OVM

specifies that you want to list the information contained in the OVM segment of the user's profile.

The details that RACF lists from the OVM segment are the z/OS UNIX System Services user's:

- User identifier
- Initial directory pathname
- Program pathname
- File system root name

If there is no HOME, PROGRAM, or FSROOT information, the field name is not displayed. However, the word "NONE" appears in the listing if the UID was not specified, or if the UID was removed using the NOUID operand on the ALTUSER command.

PROXY

specifies that PROXY segment information should be listed.

The BINDPW password value will not be listed. If a BINDPW password value is defined for a user, LISTUSER will display 'YES' for the PROXY segment BINDPW attribute. If no BINDPW password value has been defined, LISTUSER will display 'NO' for the PROXY segment BINDPW attribute.

TSO

specifies that you want to list the information contained in the TSO segment of the user's profile.

The details RACF lists from the TSO segment of the user's profile are:

- The user's default account number when logging on from the TSO/E logon panel
- The destination ID for SYSOUT data sets
- The user's default HOLDCLASS
- The user's default JOBCLASS
- The user's default MSGCLASS
- The user's default SYSOUTCLASS
- The maximum region size
- The default region size
- The logon procedure name
- The unit name
- The optional user data
- The user's security label
- The default command to be run during the TSO/E logon

If there is no information in the fields of the TSO segment, the field name is not displayed (with the exception of SIZE, MAXSIZE, and USERDATA).

WORKATTR

specifies that you want to list the information contained in the WORKATTR segment of the user's profile.

The details RACF lists for the distribution information from the user's WORKATTR segment are:

- The name of the user (WANAME)
- The building (WABLDG)
- The department (WADEPT)
- The room (WAROOM)
- Up to four additional lines of output distribution information (WAADDR1-4)
- An account number for APPC/MVS processing (WAACCNT)

Examples

Table 32. LISTUSER Examples

Example 1	<i>Operation</i>	User DAF0 wants to list the user attributes from the RACF segment of her user profile.
	<i>Known</i>	User DAF0 is RACF-defined. User DAF0 wants to issue the command as a RACF TSO command.
	<i>Command</i>	LISTUSER
	<i>Defaults</i>	DAF0 (userid)
Example 2	<i>Output</i>	See Figure 18.
	<i>Operation</i>	User CALTMANN wants to list the user attributes from the RACF segment of profiles for users IBMUSER, CALTMANN, and DAF0.
	<i>Known</i>	User CALTMANN has the SPECIAL and AUDITOR attributes. User CALTMANN wants to issue the command as a RACF TSO command.
	<i>Command</i>	LISTUSER (IBMUSER CALTMANN DAF0)
	<i>Defaults</i>	None
	<i>Output</i>	See Figure 19.

LISTUSER

Table 32. LISTUSER Examples (continued)

Example 3	<i>Operation</i>	User ADM1 wants to list the user attributes from the RACF segment and TSO segment of the profile for user DAF0.
	<i>Known</i>	User ADM1 has the SPECIAL and AUDITOR attributes.
		User DAF0 is defined to RACF with authority to use TSO.
		User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	LISTUSER DAF0 TSO
	<i>Defaults</i>	None
Example 4	<i>Output</i>	See Figure 20.
	<i>Operation</i>	User ADM1 wants to list the user attributes from only the TSO segment of the profile for user DAF0.
	<i>Known</i>	User ADM1 has the SPECIAL and AUDITOR attributes.
		User DAF0 is defined to RACF with authority to use TSO.
		User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	LISTUSER DAF0 NORACF TSO
Example 5	<i>Defaults</i>	None
	<i>Output</i>	See Figure 21.
	<i>Operation</i>	User ADM1 wants to list the user attributes from the RACF segment and DFP segment of the profile for user DAF0.
	<i>Known</i>	User ADM1 has the SPECIAL and AUDITOR attributes.
		User DAF0 is defined to RACF and DAF0's profile contains a DFP segment.
		User ADM1 wants to issue the command as a RACF TSO command.
Example 6	<i>Command</i>	LISTUSER DAF0 DFP
	<i>Defaults</i>	None
	<i>Output</i>	See Figure 22.
	<i>Operation</i>	User ADM1 wants to list the user attributes from only the DFP segment of the profile for user DAF0.
	<i>Known</i>	User ADM1 has the SPECIAL and AUDITOR attributes.
		User DAF0 is defined to RACF and DAF0's profile contains a DFP segment.
Example 7		User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	LISTUSER DAF0 NORACF DFP
	<i>Defaults</i>	None
	<i>Output</i>	See Figure 23.
	<i>Operation</i>	User ADM1 wants to list the user attributes from only the CICS segment of the profile for user DAF0.
	<i>Known</i>	User ADM1 has the SPECIAL and AUDITOR attributes.
		User DAF0 is defined to RACF and DAF0's profile contains a CICS segment.
		User is running a level of CICS prior to CICS 4.1.
		User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	LISTUSER DAF0 NORACF CICS
	<i>Defaults</i>	None
	<i>Output</i>	See Figure 24.

Table 32. LISTUSER Examples (continued)

Example 8	<i>Operation</i>	User ADM1 wants to list the user attributes from only the CICS segment of the profile for user DAF0.
	<i>Known</i>	User ADM1 has the SPECIAL and AUDITOR attributes.
		User DAF0 is defined to RACF and DAF0's profile contains a CICS segment.
		User is running CICS 4.1 or later.
Example 9		User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	LISTUSER DAF0 NORACF CICS
	<i>Defaults</i>	None
	<i>Output</i>	See Figure 25.
Example 10	<i>Operation</i>	User ADM1 wants to list the user attributes from only the LANGUAGE segment of the profile for user DAF0.
	<i>Known</i>	User ADM1 has the SPECIAL and AUDITOR attributes.
		User DAF0 is defined to RACF and DAF0's profile has American English (language code ENU) defined as her primary language and German (language code DEU) defined as her secondary language.
		User ADM1 wants to issue the command as a RACF TSO command.
Example 11	<i>Command</i>	LISTUSER DAF0 NORACF LANGUAGE
	<i>Defaults</i>	None
	<i>Output</i>	See Figure 26.
	<i>Operation</i>	User ADM1 wants to list the user attributes from only the OPERPARM segment of the profile for user DAF0.
Example 11	<i>Known</i>	User ADM1 has the SPECIAL and AUDITOR attributes.
		User DAF0 is defined to RACF and DAF0's profile contains an OPERPARM segment.
		User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	LISTUSER DAF0 NORACF OPERPARM
Example 11	<i>Defaults</i>	None
	<i>Output</i>	See Figure 27.
	<i>Operation</i>	User ADM1 wants to list the user attributes from the OMVS segment of the profile for user CSMITH.
	<i>Known</i>	User ADM1 has the SPECIAL attribute.
Example 11		User CSMITH is defined to RACF and CSMITH's profile contains an OMVS segment.
		User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	LISTUSER CSMITH OMVS NORACF
	<i>Defaults</i>	None
Example 11	<i>Output</i>	See Figure 28.

LISTUSER

Table 32. LISTUSER Examples (continued)

Example 12	<i>Operation</i>	User ADM1 wants to list the user attributes from the OMVS segment of the profile for user CSMITH.
	<i>Known</i>	User ADM1 has the SPECIAL attribute.
		User CSMITH is defined to RACF and CSMITH's profile contains an OMVS segment, but there was no value specified for HOME or PROGRAM in the OMVS segment for this profile. Defaults were used.
		User ADM1 wants to issue the command as a RACF TSO command.
Example 13	Note:	If the user also has no user limits because the defaults were taken, CPUTIMEMAX, ASSIZEMAX, FILEPROC MAX, PROCUSERMAX, THREADSMAX, and MMAPAREAMAX will display NONE as their value.
	<i>Command</i>	LISTUSER CSMITH OMVS NORACF
	<i>Defaults</i>	None
	<i>Output</i>	See Figure 29.
Example 14	<i>Operation</i>	User ADM1 wants to list the DCE segment for user CSMITH.
	<i>Known</i>	User ADM1 has the SPECIAL attribute.
	<i>Command</i>	LISTUSER CSMITH NORACF DCE
	<i>Defaults</i>	None
Example 15	<i>Output</i>	See Figure 30.
	<i>Operation</i>	A security administrator lists the KERB segment of the altered RACF user profile for RONTOMS.
	<i>Known</i>	The administrator wants to list the information contained in the KERB segment of the altered RACF user profile.
	<i>Command</i>	LISTUSER RONTOMS NORACF KERB
Example 16	<i>Defaults</i>	None
	<i>Output</i>	See Figure 31.
	<i>Operation</i>	A security administrator lists the PROXY segment of the altered RACF user profile for MRSERVER.
	<i>Known</i>	The administrator wants to list the information contained in the PROXY segment of the altered RACF user profile.
Example 17	<i>Command</i>	LISTUSER MRSERVER PROXY NORACF
	<i>Defaults</i>	None
	<i>Output</i>	See Figure 32.

```

LISTUSER
USER=DAF0      NAME=D.M.BROWN  OWNER=IBMUSER  CREATED=85.228
DEFAULT-GROUP=RESEARCH  PASSDATE=85.220  PASS-INTERVAL= 30
ATTRIBUTES=ADSP
REVOKE DATE=NONE  RESUME DATE=NONE
LAST-ACCESS=85.228/13:31:11
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED      (DAYS)          (TIME)
-----
ANYDAY                                ANYTIME
GROUP=RESEARCH AUTH=JOIN  CONNECT-OWNER=IBMUSER  CONNECT-DATE=85.228
CONNECTS=    01  UACC=READ  LAST-CONNECT=85.228/13:31:11
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE  RESUME DATE=NONE
GROUP=PAYROLLB AUTH=CREATE  CONNECT-OWNER=IBMUSER  CONNECT-DATE=85.228
CONNECTS=    00  UACC=READ  LAST-CONNECT=UNKNOWN
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE  RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
NONE SPECIFIED

```

Figure 18. Example 1: Output for LISTUSER

```

LISTUSER (IBMUSER CALTMANN DAF0)
USER=IBMUSER  NAME=G. SMITH  OWNER=IBMUSER  CREATED=84.263
DEFAULT-GROUP=SYS1  PASSDATE=85.104  PASS-INTERVAL=N/A
ATTRIBUTES=SPECIAL OPERATIONS
ATTRIBUTES=AUDITOR
REVOKE DATE=NONE  RESUME DATE=NONE
LAST-ACCESS=85.146/15:45:23
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED      (DAYS)          (TIME)
-----
ANYDAY                                ANYTIME
GROUP=SYS1      AUTH=JOIN  CONNECT-OWNER=IBMUSER  CONNECT-DATE=84.263
CONNECTS=    456  UACC=READ  LAST-CONNECT=85.146/15:45:23
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE  RESUME DATE=NONE
GROUP=VSAMDSET AUTH=JOIN  CONNECT-OWNER=IBMUSER  CONNECT-DATE=84.263
CONNECTS=    00  UACC=NONE  LAST-CONNECT=UNKNOWN
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE  RESUME DATE=NONE
GROUP=SYSCTLG  AUTH=JOIN  CONNECT-OWNER=IBMUSER  CONNECT-DATE=84.263
CONNECTS=    00  UACC=READ  LAST-CONNECT=UNKNOWN
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE  RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED

```

Figure 19. Example 2: Output for LISTUSER (IBMUSER CALTMAN DAF0) (Part 1 of 2)

LISTUSER

```

USER=CALTMANN NAME=C. ALTMANN OWNER=IBMUSER CREATED=85.144
DEFAULT-GROUP=RESEARCH PASSDATE=00.000PASS-INTERVAL=254
ATTRIBUTES=SPECIAL
ATTRIBUTES=AUDITOR
REVOKE DATE=NONE RESUME DATE=NONE
LAST-ACCESS=85.146/16:16:14
CLASS AUTHORIZATIONS=USER
NO-INSTALLATION-DATA
MODEL-NAME=ALLENA
LOGON ALLOWED (DAYS) (TIME)
-----
ANYDAY ANYTIME
GROUP=RESEARCH AUTH=JOIN CONNECT-OWNER=IBMUSER CONNECT-DATE=85.144
CONNECTS= 01 UACC=READ LAST-CONNECT=85.146/16:16:14
CONNECT ATTRIBUTES=OPERATIONS
REVOKE DATE=NONE RESUME DATE=NONE
GROUP=VSAMDSET AUTH=CREATE CONNECT-OWNER=IBMUSER CONNECT-DATE=85.144
CONNECTS= 00 UACC=READ LAST-CONNECT=UNKNOWN
CONNECT ATTRIBUTES=OPERATIONS
REVOKE DATE=NONE RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED
USER=DAF0 NAME=D.M.BROWN OWNER=IBMUSER CREATED=85.144
DEFAULT-GROUP=RESEARCH PASSDATE=00.000 PASS-INTERVAL= 254
ATTRIBUTES=ADSP
REVOKE DATE=NONE RESUME DATE=NONE
LAST-ACCESS=85.146/15:11:31
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED (DAYS) (TIME)
-----
ANYDAY ANYTIME
GROUP=RESEARCH AUTH=JOIN CONNECT-OWNER=IBMUSER CONNECT-DATE=85.144
CONNECTS= 02 UACC=READ LAST-CONNECT=85.146/15:11:31
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED

```

Figure 19. Example 2: Output for LISTUSER (IBMUSER CALTMAN DAF0) (Part 2 of 2)

```

LISTUSER DAF0 TSO
USER=DAF0      NAME=D.M.BROWN  OWNER=IBMUSER  CREATED=85.228
DEFAULT-GROUP=RESEARCH  PASSDATE=85.220  PASS-INTERVAL= 30
ATTRIBUTES=ADSP
REVOKE DATE=NONE  RESUME DATE=NONE
LAST-ACCESS=85.228/13:31:11
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED      (DAYS)          (TIME)
-----
ANYDAY                                ANYTIME
GROUP=RESEARCH AUTH=JOIN  CONNECT-OWNER=IBMUSER  CONNECT-DATE=85.228
CONNECTS= 01 UACC=READ  LAST-CONNECT=85.228/13:31:11
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE  RESUME DATE=NONE
GROUP=PAYROLLB AUTH=CREATE  CONNECT-OWNER=IBMUSER  CONNECT-DATE=85.228
CONNECTS= 00 UACC=READ  LAST-CONNECT=UNKNOWN
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE  RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED
TSO INFORMATION
ACCTNUM= P00F1V
HOLDCLASS= H
JOBCLASS= I
MSGCLASS= A
PROC= V0LOGON
SIZE= 00001024
MAXSIZE= 00002048
SYSOUTCLASS= A
UNIT= SYSDA
USERDATA= 0000

```

Figure 20. Example 3: Output for LISTUSER DAF0 TSO

```

LISTUSER DAF0 NORACF TSO
USER=DAF0
TSO INFORMATION
ACCTNUM= P00F1V
HOLDCLASS= H
JOBCLASS= I
MSGCLASS= A
PROC= V0LOGON
SIZE= 00001024
MAXSIZE= 00002048
SYSOUTCLASS= A
UNIT= SYSDA
USERDATA= 0000

```

Figure 21. Example 4: Output for LISTUSER NORACF TSO

LISTUSER

```
LISTUSER DAF0 DFP
USER=DAF0      NAME=D.M.BROWN  OWNER=IBMUSER  CREATED=85.228
DEFAULT-GROUP=RESEARCH  PASSDATE=85.220  PASS-INTERVAL= 30
ATTRIBUTES=ADSP
REVOKE DATE=NONE  RESUME DATE=NONE
LAST-ACCESS=85.228/13:31:11
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED      (DAYS)          (TIME)
-----
ANYDAY                                ANYTIME
GROUP=RESEARCH AUTH=JOIN  CONNECT-OWNER=IBMUSER  CONNECT-DATE=85.228
CONNECTS= 01 UACC=READ  LAST-CONNECT=85.228/13:31:11
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE  RESUME DATE=NONE
GROUP=PAYROLLB AUTH=CREATE  CONNECT-OWNER=IBMUSER  CONNECT-DATE=85.228
CONNECTS= 00 UACC=READ  LAST-CONNECT=UNKNOWN
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE  RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED
DFP INFORMATION
MGMTCLAS= DFP5MGMT
STORCLAS= DFP5STOR
DATACLAS= DFP5DATA
DATAAPPL= DFP5APPL
```

Figure 22. Example 5: Output for LISTUSER DAF0 DFP

```
LISTUSER DAF0 NORACF DFP
USER=DAF0
DFP INFORMATION
MGMTCLAS= DFP5MGMT
STORCLAS= DFP5STOR
DATACLAS= DFP5DATA
DATAAPPL= DFP5APPL
```

Figure 23. Example 6: Output for LISTUSER DAF0 NORACF DFP

```
LISTUSER DAF0 NORACF CICS
USER=TEST
CICS INFORMATION
-----
OPCLASS= 001
OPIDENT= ID2
OPPRTY= 00010
TIMEOUT= 005
XRFSOFF= NOFORCE
```

Figure 24. Example 7: Output for LISTUSER DAF0 NORACF CICS (prior to CICS 4.1)

```

LISTUSER DAF0 NORACF CICS
USER=TEST
CICS INFORMATION
-----
OPCLASS= 001
OPIDENT= ID2
OPPRTY= 00010
TIMEOUT= 02:30 (HH:MM)
XRFSOFF= NOFORCE

```

Figure 25. Example 8: Output for LISTUSER DAF0 NORACF CICS (CICS 4.1)

```

LISTUSER DAF0 NORACF LANGUAGE
USER=DAF0
LANGUAGE INFORMATION
-----
PRIMARY LANGUAGE: ENU
SECONDARY LANGUAGE: DEU
READY

```

Figure 26. Example 9: Output for LISTUSER DAF0 NORACF LANGUAGE

```

LU DAF0 NORACF OPERPARM
USER=DAF0
OPERPARM INFORMATION
-----
STORAGE= 00002
AUTH= IO
ROUTCODE= ALL
LEVEL= ALL
MFORM= T J M
MONITOR= JOBNAMEST SESST
MIGID= YES
DOM= NORMAL
KEY= MCS2
CMDSYS= SYS1
MSCOPE= *ALL
UD= YES
READY

```

Note: With the exception of the STORAGE operand, if a field has no value in the OPERPARM segment, no value appears for the field in the listing. If there is an OPERPARM segment and the storage is not specified, 00000 appears in the listing. When an extended MCS console session is established, the values for STORAGE is 1.

Figure 27. Example 10: Output for LISTUSER DAF0 NORACF OPERPARM

LISTUSER

```
LISTUSER CSMITH OMVS NORACF
USER = CSMITH
OMVS INFORMATION
-----
UID= 0000000024
HOME= /u/CSMITH
PROGRAM= /u/CSMITH/bin/myshe11
CPUTIMEMAX= 0010000000
ASSIZEMAX= NONE
FILEPROCMAx= 0000050000
PROUSERMAX= NONE
THREADSMAX= NONE
MMAPAREAMAX= 0016777216
```

Figure 28. Example 11: Output for listing OMVS user information

```
LISTUSER CSMITH OMVS NORACF
USER=CSMITH
OMVS INFORMATION
-----
UID= 0000000024
CPUTIMEMAX= NONE
ASSIZEMAX= NONE
FILEPROCMAx= NONE
PROUSERMAX=NONE
THREADSMAX= NONE
MMAPAREAMAX= NONE
```

Figure 29. Example 12: Output for LISTUSER CSMITH OMVS NORACF (Using Defaults)

```
LISTUSER CSMITH NORACF DCE
USER=CSMITH
DCE INFORMATION
-----
UUID= 004386ea-ebb6-1ec3-bcae-10005ac90feb
DCENAME= charlie
HOME CELL UUID= 003456aab-ecb7-7de3-ebda-95531ed63dae
HOME CELL= ../../hootie.scarol.ibm.com
DCE AUTOLOGIN= NO
```

Figure 30. Example 13: Output for LISTUSER CSMITH NORACF DCE

```
LISTUSER RONTOMS NORACF KERB
USER=RONTOMS
KERB INFORMATION
-----
KERBNAME=KerberizedUser
KEY VERSION=001
```

Figure 31. Example 14: Output for LISTUSER RONTOMS NORACF KERB

USER=MRSERVER

PROXY INFORMATION

LDAPHOST= LDAP://SOME.LDAP.HOST:389

BINDDN= cn=Joe User,ou=Poughkeepsie,o=IBM,c=US

BINDPW= YES

Figure 32. Example 15: Output for LISTUSER MRSERVER PROXY NORACF

USER=MRSERVER

EIM INFORMATION

LDAPPROF= EIMDOMAINALOOKUP

Figure 33. Example 16: Output for LISTUSER MRSERVER EIM NORACF

PASSWORD (Specify User Password)

Purpose

Use the PASSWORD command to:

- Change your current password to a specified value,
- Change a user's password interval (the number of days that a password remains valid),
- Specify a password that never expires, or
- Reset a user's password to a known default value.

The PASSWORD command allows you to change your password at any time. It also allows you to change the number of days that a password is valid or to specify that a current password is valid indefinitely.

If you use the PASSWORD command to change your own password and you have user ID associations with password synchronization defined, the password is synchronized. However, if you use the PASSWORD command to change another user's password, the password is not synchronized.

Attention:

- When the PASSWORD command is issued from ISPF, the TSO command buffer (including password data) is written to the ISPLLOG data set. As a result, you should not issue this command from ISPF or you must control the ISPLLOG data set carefully.
- If the PASSWORD command is issued as a RACF operator command, the command and the password data is written to the system log. Therefore, use of PASSWORD as a RACF operator command should either be controlled or you should issue the command as a TSO command.

Issuing Options

The following table identifies the eligible options for issuing the PASSWORD command:

Table 33. How the PASSWORD Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	Yes	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, "RACF TSO commands" on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, "RACF operator commands" on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To change the installation password interval, see "SETROPTS (Set RACF Options)" on page 435.

- To establish password synchronization between users, see “RACLINK (Administer User ID Associations)” on page 296.

Authorization Required

When issuing the PASSWORD command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

If you are a RACF-defined user and you are required to provide a RACF user password when entering the system, you can change your own password or your password change interval.

To reset another user's password to the user's default value, one of the following conditions must be true:

- You have the SPECIAL attribute
- The user's profile is within the scope of a group in which you have the group-SPECIAL attribute
- You are the owner of the user's profile.

To change another user's password interval, or to set a password that never expires, you must have the SPECIAL attribute, or the user's profile must be within the scope of a group in which you have the group-SPECIAL attribute.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the PASSWORD command is:

```
[subsystem-prefix]{PASSWORD | PW}
                        [ AT( [ node ] .userid ...) | ONLYAT( [ node ] .userid ...) ]
                        [ INTERVAL(change-interval) | NOINTERVAL ]
                        [ PASSWORD(current-password new-password) ]
                        [ USER(userid ...) ]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the

PASSWORD

command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([*node*].*userid* ...)

specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT([*node*].*userid* ...)

specifies that the command is to be directed only to the node specified by *node* where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed only to the local node.

INTERVAL | NOINTERVAL

INTERVAL(*change-interval*)

change-interval indicates the number of days during which a password remains valid; the range is from 1 through 254 days.

The value you specify here cannot exceed the value, if any, that your installation has specified using the INTERVAL operand on the SETROPTS command. The initial system default after RACF initialization is 30 days.

If you specify INTERVAL on the PASSWORD command without a *change-interval* value, RACF uses the installation-specified maximum.

To specify INTERVAL with USER, you must have the SPECIAL attribute, or the user profile must be within the scope of a group in which you have the group-SPECIAL attribute.

If you specify the interval incorrectly, RACF ignores this operand.

NOINTERVAL

sets a password that never expires. To specify NOINTERVAL, you must have the SPECIAL attribute, or the user profile must be within the scope of a group in which you have the group-SPECIAL attribute.

Specifying NOINTERVAL without USER defines your own password as a password that never expires.

You can use INTERVAL at any time to reinstate an expiration interval for a password previously defined with NOINTERVAL.

PASSWORD(*current-password* *new-password*)

specifies your current password and the new one you want. If you enter only the PASSWORD operand, you are prompted so you can enter the current and new passwords in print inhibit mode.

The current and new passwords must have different values. If you specify your current password incorrectly, RACF notifies you and ignores the PASSWORD operand.

You can use the PASSWORD operand to change your own password at any time.

RACF ignores this operand when you specify the USER operand.

USER(userid ...)

specifies one or more users whose passwords are to be reset. Unless you also specify INTERVAL or NOINTERVAL, each user's password is set to the user's respective default group name and the password is set expired.

Notes:

1. To change your own password, use the PASSWORD operand, not the USER operand.
2. Specifying USER with your own user ID resets your password to the name of your default group, and sets the password as expired, unless you also specify INTERVAL or NOINTERVAL.
If you specify USER with the PASSWORD operand, the PASSWORD operand is ignored.
3. When a password is changed for its user, RACF makes sure the proposed password is not the same as the current password or in the password history list. If the proposed password does not match one of these passwords, the current password is added to the password history list, and the proposed password is made current.

Examples

Table 34. PASSWORD Examples

Example 1

Operation User AEH0 wants to change his password from XY262 to YZ344 and increase his change interval to 60 days.

Known User AEH0 is RACF-defined.

The maximum installation change-interval is at least 60 days.

User AEH0 wants to issue the command as a RACF TSO command.

Command PASSWORD PASSWORD(XY262 YZ344) INTERVAL(60)

Example 2

Operation User ADM1 wants to reset the passwords for users CD0 and DAF0 to the names of their default group.

Known User ADM1 has the group-SPECIAL attribute in group PAYROLL. Group PAYROLL is the owning group of users CD0 and DAF0.

Users CD0 and DAF0 are RACF-defined.

User ADM1 wants to issue the command as a RACF TSO command.

Command PASSWORD USER(CD0 DAF0)

PASSWORD

Table 34. PASSWORD Examples (continued)

Example 3

Operation User ADM1 wants to set a password that never expires for user CD2. User ADM1 wants to direct the command to run under the authority of CHERYLB at node ALBANY and prohibit the command from being automatically directed to other nodes.

Known Users ADM1 and CHERYLB at ALBANY have the SPECIAL attribute.

User CD2 is RACF-defined on node ALBANY. Users ADM1 and CHERYLB at ALBANY have an already established user ID association.

User ADM1 wants to issue the command as a RACF TSO command.

Command PASSWORD USER(CD2) NOINTERVAL ONLYAT(ALBANY.CHERYLB)

Results The command is only processed at node ALBANY and not automatically directed to any other nodes in the RRSF configuration.

Example 4

Operation Bob wants to change his password from pass1 to word1 on both his user IDs. His user IDs are BOB1 on MVS01 and BOB2 on MVS02.

Known Bob has a peer user ID association with password synchronization established between his two user IDs. Bob wants to issue the command as a RACF TSO command from MVS01.

Command PASSWORD PASSWORD(pass1 word1)

Results The command is processed on MVS01 and the password is changed for user ID BOB1. The password is also changed for user ID BOB2 at MVS02.

PERMIT (Maintain Resource Access Lists)

Purpose

Use the PERMIT command to maintain the lists of users and groups authorized to access a particular resource. RACF provides two types of access lists: standard and conditional.

Standard Access List: The standard access list includes the user IDs and group names authorized to access the resource and the level of access granted to each.

Conditional Access List: The conditional access list includes the user IDs and group names authorized to access the resource and the level of access granted to each when a certain condition is met. The conditions that can be specified are:

1. The name of the program the user must be executing,
2. The name of the terminal by which the user entered the system,
3. The name of the JES input device through which the user entered the system,
4. The name of the system console from which the request was originated,
5. The name of the APPC partner LU (logical unit) from which the transaction program originated, and
6. The system identifier (SMFID) of the system on which the user is loading the controlled program.

RACF considers the conditional access list if one of the following is true:

- The class specified in the condition is active (for the TERMINAL, JESINPUT, CONSOLE, or APPCPORT conditions).
- The RACF program control facility is active (for the PROGRAM or the SYSID condition). The RACF program control facility is activated by your installation using SETROPTS WHEN(PROGRAM) command.

If one of the criteria above is met, RACF uses both the standard and conditional access lists when it checks a user's authority to access a resource; otherwise RACF uses only the standard access list. For more information on conditional access lists or program control, see *z/OS Security Server RACF Security Administrator's Guide*.

You can maintain either the standard access list or the conditional access list with a single PERMIT command. Changing both requires you to issue PERMIT twice, with one exception. You can change individual names in one access list and copy the other access list from another profile on one PERMIT command.

Using PERMIT, you can make the following changes to either a standard access list or a conditional access list:

- Give authority to access a discrete or generic resource profile to specific RACF-defined users or groups
- Remove authority to access a discrete or generic resource profile from specific users or groups
- Change the level of access authority to a discrete or generic resource profile for specific users or groups
- Copy the list of authorized users from one discrete or generic resource profile to another profile of either type and modify the new list as you require
- Delete an existing access list.

PERMIT

Using PERMIT to modify an automatic TAPEVOL profile changes the profile to nonautomatic. For more information about TAPEVOL profiles, see *z/OS Security Server RACF Security Administrator's Guide*.

To have changes take effect after updating a user's access to a generic profile, one of the following steps is required:

- If the command was issued for a data set profile, the user of the data set issues the LISTDSD command:

```
LISTDSD DA(data-set-protected-by-the-profile) GENERIC
```

Note: Use the data set name, not the profile name.

- The security administrator issues the SETROPTS command:

```
SETROPTS GENERIC(class-name) REFRESH
```

See SETROPTS command for authorization requirements.

- The user of the data set or resource logs off and logs on again.

Note: For more information, refer to *z/OS Security Server RACF Security Administrator's Guide*.

Issuing Options

The following table identifies the eligible options for issuing the PERMIT command:

Table 35. How the PERMIT Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	Yes	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To specify the UACC for a data set profile, see “ADDSD (Add Data Set Profile)” on page 33 (when creating a new profile), or “ALTDSD (Alter Data Set Profile)” on page 90 (to change an existing profile).
- To specify the UACC for a general resource (such as a terminal), see “RDEFINE (Define General Resource Profile)” on page 337 (when creating a new profile), or 303 (to change an existing profile).
- To obtain a list of profiles, see “SEARCH (Search RACF Database)” on page 408.
- To list a data set profile, see “LISTDSD (List Data Set Profile)” on page 200.
- To list a general resource profile, see “RLIST (List General Resource Profile)” on page 382. (General resources include terminals, and other resources defined in the class descriptor table.)

Authorization Required

When issuing the PERMIT command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

To perform any of the PERMIT functions, you must have sufficient authority over the resource. RACF makes the following checks until one of the conditions is met:

- You have the SPECIAL attribute.
- The profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You are the owner of the resource.
- If the resource belongs to the DATASET class, the high-level qualifier of the profile name (or the qualifier supplied by the naming conventions routine or a command installation exit) is your user ID.
- If the resource belongs to the DATASET class, you must be the current owner of the profile or have the SPECIAL attribute, or the profile must be within the scope of a group in which you have the group-SPECIAL attribute.
- If the profile is in the FILE or DIRECTORY class, the second qualifier of the profile name is your user ID.

For discrete profiles only:

- You are on the standard access list for the resource and you have ALTER authority.
- Your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is on the standard access list and has ALTER authority.
- The universal access authority is ALTER.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

When you are copying a list of authorized users from one resource profile to another, you must have sufficient authority, as described in the preceding list, to both of the resources.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the PERMIT command is:

```
[subsystem-prefix]{PERMIT | PE}
                                profile-name-1
                                [ ACCESS(access-authority) | DELETE ]
                                [ AT( [ node ] .userid ...) | ONLYAT( [ node ] .userid ...) ]
                                [ CLASS(profile-name-1-class) ]
                                [ FCLASS(profile-name-2-class) ]
```

PERMIT

```
[ FGENERIC ]  
[ FROM(profile-name-2) ]  
[ FVOLUME(volume-serial) ]  
[ GENERIC ]  
[ ID( {name ... !*} ) ]  
[ RESET [ (ALL | STANDARD | WHEN) ]  
[ VOLUME(volume-serial) ]  
[ WHEN(  
  [ APPCPORT( {partner-luname ...! *} ) ]  
  [ CONSOLE( {console-id ...! *} ) ]  
  [ JESINPUT( {JES-input-device-name ...! *} ) ]  
  [ PROGRAM( {program-name ...! *} ) ]  
  [ SYSID( {system-identifier ...! *} ) ]  
  [ TERMINAL( {terminal-id ...! *} ) ]  
  ) ]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

profile-name-1

specifies the name of an existing discrete or generic profile whose access list you want to modify. You can specify only one profile.

This operand is required and must be the first operand following PERMIT.

If the name specified is a tape volume serial number that is a member of a tape volume set, the authorization assigned by this command applies to all the volumes in the volume set.

If the profile does not belong to the DATASET class, you must also specify CLASS.

Mixed case profile names are accepted and preserved when CLASS refers to a class defined in the class descriptor table with CASE=ASIS.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([*node*].*userid* ...)

specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT(*[node].userid ...*)

specifies that the command is to be directed only to the node specified by *node* where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed only to the local node.

ACCESS | DELETE

ACCESS(*access-authority*)

specifies the access authority you want to associate with the names that you identify on the ID operand. RACF sets the access authority in the standard access list.

If you specify WHEN, RACF sets the access authority in the conditional access list.

The valid access authorities are NONE, EXECUTE (for DATASET or PROGRAM class only), READ, UPDATE, CONTROL, and ALTER. If you need more information, see *z/OS Security Server RACF Security Administrator's Guide*.

If you specify ACCESS and omit *access-authority*, the default value is ACCESS(READ).

If you specify the ID operand and omit both ACCESS and DELETE, the default value is ACCESS(READ).

If you specify both ACCESS and DELETE, RACF uses the last operand you specify.

DELETE

specifies that you are removing the names you identify on the ID operand from an access list for the resource. RACF deletes the names from the standard access list.

If you specify WHEN, RACF deletes the names from the conditional access list.

If you specify the ID operand and omit both ACCESS and DELETE, the default value is ACCESS(READ).

If you specify both ACCESS and DELETE, RACF uses the last operand you specify.

CLASS(*profile-name-1-class*)

specifies the name of the class to which *profile-name-1* belongs. The valid class names are DATASET and those classes defined in the class descriptor table. For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, "Description of RACF Classes" on page 507.

If you omit CLASS, the default is DATASET.

FCLASS(*profile-name-2-class*)

specifies the name of the class to which *profile-name-2* belongs. The valid class names are DATASET and those classes defined in the class descriptor table. For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, "Description of RACF Classes" on page 507.

If you specify FROM and omit FCLASS, RACF assumes that the class for *profile-name-2* is same as the class for *profile-name-1*. This operand is valid only when you also specify the FROM operand; otherwise, RACF ignores it.

PERMIT

FGENERIC

specifies that RACF is to treat *profile-name-2* as a generic name, even if it is fully-qualified (meaning that it does not contain any generic characters). This operand is only needed if *profile-name-2* is a DATASET profile.

FROM(*profile-name-2*)

specifies the name of the existing discrete or generic profile that contains the access lists RACF is to copy as the access lists for *profile-name-1*. If you specify FROM and omit FCLASS, RACF assumes that *profile-name-2* is the name of a profile in the same class as *profile-name-1*.

Mixed case profile names are accepted and preserved when FCLASS refers to a class defined in the class descriptor table with CASE=ASIS.

If *profile-name-2* contains a standard access list, RACF copies it to the profile you are changing. If *profile-name-2* contains a conditional access list, RACF copies it to the profile you are changing.

RACF modifies the access list for *profile-name-1* as follows:

- Authorizations for *profile-name-2* are added to the access list for *profile-name-1*
- If a group or user appears in both lists, RACF uses the authorization granted in *profile-name-1*
- If you specify a group or user on the ID operand and that group or user also appears in the *profile-name-2* access list, RACF uses the authorization granted on the ID operand.

To specify FROM, you must have sufficient authority to both *profile-name-1* and *profile-name-2*, as described under “Authorization Required” on page 249.

FVOLUME(*volume-serial*)

specifies the volume RACF is to use to locate *profile-name-2*. This is the volume on which the non-VSAM DASD data set, the tape data set, or the catalog for the VSAM data set resides.

If you specify FVOLUME and RACF does not find *profile-name-2* on that volume, the command fails. If you omit this operand and *profile-name-2* appears more than once in the RACF data set, the command fails.

FVOLUME is valid only when FCLASS either specifies or defaults to DATASET and when *profile-name-2* specifies a discrete profile. Otherwise, RACF ignores FVOLUME.

GENERIC

specifies that RACF is to treat *profile-name-1* as a generic name, even if it does not contain any generic characters. This operand is only needed if *profile-name-1* is a DATASET profile.

ID(*name ...I**)

specifies the user IDs and group names of RACF-defined users or groups whose authority to access the resource you are giving, removing, or changing. If you omit this operand, RACF ignores the ACCESS and DELETE operands.

ID(*) can be used with standard or conditional access lists. You might specify ID(*) with a conditional access list, as follows:

```
PERMIT 'resource' ID(*) WHEN(PROGRAM(XYZ)) ACCESS(READ)
```

This command, depending on other environmental factors, may allow all RACF-defined users and groups READ access to the specified data set when executing program XYZ. RACF grants access to the data set, using the conditional access list, with the authority you specify on the ACCESS operand.

The value specified with ACCESS is used only if no more specific values are found. If you do not specify the ACCESS operand, or if you specify ACCESS without an access authority, RACF uses a default value of ACCESS(READ). See *z/OS Security Server RACF Security Administrator's Guide* for more information on program access to data sets.

For profiles in the FIELD class, you may also specify the value &RACUID for the *name* variable with the ID operand on the PERMIT command. When you enter this value on the PERMIT command, you allow all users access to the specified field or segment of their own user profiles.

RESET [(ALL | STANDARD | WHEN)]

RESET | RESET(ALL)

specifies that RACF is to delete from the profile both the entire current standard access list and the entire current conditional access list.

RACF deletes both access lists before it processes any operands (ID and ACCESS or FROM) that create new entries in an access list. If you delete both access lists and specify FROM when *profile-name-2* contains two access lists, the PERMIT command copies both access lists to *profile-name-1*. In any other situation, you cannot, on one PERMIT command, add entries to both access lists.

If you specify RESET or RESET(ALL), add entries, and omit WHEN, RACF deletes both access lists, then adds entries to the standard access list.

If you specify RESET or RESET(ALL), add entries, and specify WHEN, RACF deletes both access lists, then adds entries to the conditional access list.

For profiles that include two access lists, use RESET and RESET(ALL) carefully. Unless you are copying both lists from another profile, it is a good practice to use RESET(STANDARD) to maintain the standard access list and RESET(WHEN) to maintain the conditional access list.

RESET(STANDARD)

specifies that RACF is to delete the entire current standard access list from the profile.

If you specify RESET(STANDARD) with ID and ACCESS or with FROM, RACF deletes the current standard access list from the profile before it adds the new names.

If you specify RESET(STANDARD) with ID and DELETE, RACF ignores RESET(STANDARD) and deletes only the names that you specify.

If you specify RESET(STANDARD) without ID and ACCESS, or without FROM, the resulting standard access list is empty. An empty standard access list means that, for a general resource or a group data set profile, you must be the owner or have the SPECIAL attribute, or the profile must be within the scope of a group in which you have the group-SPECIAL attribute, in order to update the access list again.

For a DATASET profile, an empty conditional access list means that no users or groups can access the data set by executing a program.

RESET(WHEN)

specifies that RACF is to delete the entire current conditional access list from the profile.

PERMIT

If you specify RESET(WHEN) with ID and ACCESS or with FROM, RACF deletes the current conditional access list from the profile before it adds the new names.

If you specify RESET(WHEN) with ID, DELETE, and WHEN, RACF ignores RESET(WHEN) and deletes only the names that you specify.

If you specify RESET(WHEN) without ID and ACCESS, or without FROM, the resulting conditional access list is empty.

VOLUME(*volume-serial*)

specifies the volume on which the tape data set, the non-VSAM DASD data set, or the catalog for the VSAM data set resides.

If you specify VOLUME and *volume-serial* does not appear in the profile for the data set, the command fails.

If you omit VOLUME and the data set name appears more than once in the RACF data set, the command fails.

This operand is valid only for CLASS(DATASET). RACF ignores it for all other CLASS values.

If *profile-name-1* is a generic profile, RACF ignores this operand.

WHEN(APPCPORT(*partner-luname ...!**))

specifies that the indicated users or groups have the specified access authority when executing commands and jobs originating from the specified partner LU.

WHEN(APPCPORT(*)) deletes all APPCPORT entries for the specified users/groups. It is valid only with the DELETE operand.

Note: The LU name must be qualified with the network name if the installation is using the network qualified names feature on the APPC connection. For more information, refer to *z/OS Security Server RACF Security Administrator's Guide*.

WHEN(CONSOLE(*console-id ...!**))

specifies that the indicated users or groups have the specified access authority when executing commands and jobs originating from the specified system console.

WHEN(CONSOLE(*)) deletes all CONSOLE entries for the specified users/groups. It is valid only with the DELETE operand.

WHEN(JESINPUT(*JES-input-device-name ...!**))

specifies that the indicated users or groups have the specified access authority when entering the system through the specific JES input device.

WHEN(JESINPUT(*)) deletes all JESINPUT entries for the specified users/groups. It is valid only with the DELETE operand.

WHEN(PROGRAM(*program-name ...!**))

Specifies that you want to create or delete entries in the conditional access list of the specified data set. This operand applies only to resources in the data set class. WHEN(PROGRAM(*)) is valid only when specified with the DELETE option.

An entry in the conditional list allows a user or group access to the data sets protected by the profile.

For example, if you enter the following command:

```
PERMIT 'XXX.YYY' ID(SMITH) ACCESS(READ) WHEN(PROGRAM(ABC))
```

RACF allows user SMITH READ access to the data set protected by profile XXX.YYY when executing program ABC. RACF grants access, through the conditional access list, with the authority you specify on the ACCESS operand. If you do not specify the ACCESS operand, or if you specify ACCESS without an access authority, RACF uses a default value of ACCESS(READ).

See *z/OS Security Server RACF Security Administrator's Guide* more information on data set access when program control is active.

WHEN(PROGRAM) affects only users and groups specified on the ID operand; it has no effect on names copied from a standard access list in another profile (using the FROM operand). Thus, you can copy a standard access list from another profile that contains only a standard access list and add or delete names in the conditional access list on a single PERMIT command.

To delete an entry from the conditional access list of a data set profile, issue the PERMIT command as follows:

```
PERMIT 'XXX.YYY' ID(JONES) DELETE WHEN(PROGRAM(ABC))
```

When you issue this command, RACF no longer allows user JONES access to the data set protected by profile XXX.YYY when executing program ABC. If you specify WHEN(PROGRAM(*)) with DELETE, RACF deletes all program names for each user or group specified on the ID operand.

See also the description of the ID operand.

WHEN(PROGRAM(*)) deletes all PROGRAM entries for the specified users/groups. It is valid only with the DELETE operand.

WHEN(SYSID(*system-identifier* ...I*))

specifies that the indicated users or groups have the specified access authority when loading this controlled program on the specified system.

This operand applies only to resources in the PROGRAM class. The *system-identifier* is the 4 character value specified for the SID parameter of the SMFPRMxx member of SYS1.PARMLIB. See *z/OS MVS Initialization and Tuning Reference* for additional information on SMFPRMxx.

WHEN(SYSID(*)) deletes all SYSID entries for the specified users/groups. It is valid only with the DELETE operand.

WHEN(TERMINAL(*terminal-id* ...I*))

specifies that the indicated users or groups have the specific access authority when logged on to the named terminal.

WHEN(TERMINAL(*)) deletes all TERMINAL entries for the specified users/groups. It is valid only with the DELETE operand.

PERMIT

Examples

Table 36. PERMIT Examples

Example 1	<i>Operation</i>	User WJE10 wants to give UPDATE access authority to data set WJE10.DEPT2.DATA to all the users in the group RESEARCH. Data set WJE10.DEPT2.DATA is protected by a discrete profile.
	<i>Known</i>	User WJE10 and group RESEARCH are RACF-defined. Data set WJE10.DEPT2.DATA is RACF-defined. User WJE10 wants to issue the command as a RACF TSO command.
	<i>Command</i>	PERMIT 'WJE10.DEPT2.DATA' ID(RESEARCH) ACCESS(UPDATE)
	<i>Defaults</i>	CLASS(DATASET)
Example 2	<i>Operation</i>	User WRH0 wants to give all users authorized to access the data set RESEARCH.PROJ01.DATA on volume DASD22 the authority to access RESEARCH.PROJ01.DATA on volume DASD11. User WRH0 also wants to give user AEH10 READ authority to RESEARCH.PROJ01.DATA.
	<i>Known</i>	User WRH0 has ALTER access to both RESEARCH.PROJ01.DATA data sets. Both data sets are protected by discrete profiles. User WRH0 wants to issue the command as a RACF TSO command.
	<i>Command</i>	PERMIT 'RESEARCH.PROJ01.DATA' ID(AEH10) FROM('RESEARCH.PROJ01.DATA') VOLUME(DASD11) FVOLUME(DASD22)
	<i>Defaults</i>	ACCESS(READ) CLASS(DATASET) FCLASS(DATASET)
Example 3	<i>Operation</i>	User LAB2 wants to delete user MMC02's access to tape volume TAP8X.
	<i>Known</i>	User LAB2 is the owner of the profile for tape volume TAP8X. User LAB2 wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.
	<i>Command</i>	@PERMIT TAP8X CLASS(TAPEVOL) ID(MMC02) DELETE
	<i>Defaults</i>	None
Example 4	<i>Operation</i>	User ADM1 wants to delete the existing standard access list from the discrete profile protecting the data set SALES.EUROPE.ABC, then copy the standard access list from the generic profile SALES.*.ABC to the discrete profile for SALES.EUROPE.ABC. User ADM1 wants to direct the command to run under the authority of user THB11.
	<i>Known</i>	User THB11 has the SPECIAL attribute. SALES.EUROPE.ABC is in the DATASET class. User ADM1 wants to issue the command as a RACF TSO command. ADM1 and THB11 have an already established user ID association.
	<i>Command</i>	PERMIT 'SALES.EUROPE.ABC' FROM ('SALES.*.ABC') RESET(STANDARD) AT(.THB11)
	<i>Defaults</i>	CLASS (DATASET) FCLASS(DATASET)
Example 5		Command direction defaults to the local node.
	<i>Operation</i>	User ADM1 wants to replace the conditional access list in the discrete profile that protects the data set SALES.EUROPE.ABC. Two users, TH01 and TH03, are to be allowed to update the data set when executing the program named FUTURE.
	<i>Known</i>	User ADM1 has the SPECIAL attribute. Users TH01 and TH03 are defined to RACF. The program FUTURES has been defined to RACF as a controlled program. User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	PERMIT 'SALES.EUROPE.ABC' RESET(WHEN) ID(TH01 TH03) ACCESS(UPDATE) WHEN(PROGRAM(FUTURES))
	<i>Defaults</i>	CLASS(DATASET)

Table 36. PERMIT Examples (continued)

Example 6	<i>Operation</i>	User ADM1 wants to control the access of shared user IDs PUBLIC and RESELL to data sets containing sales data. All users working within the company need access to sales data along with RESELL, but PUBLIC cannot have access.
	<i>Known</i>	User ADM1 has the SPECIAL attribute. User IDs PUBLIC and RESELL have the RESTRICTED attribute. SALES RESELL.* is a generic data set with a UACC(READ).
	<i>Command</i>	PERMIT 'SALES.RESELL.*' ID(RESELL) ACCESS(READ)
	<i>Defaults</i>	None

RACDCERT (RACF Digital Certificate)

Purpose

RACDCERT is used to install and maintain digital certificates, key rings, and digital certificate mappings in RACF. RACDCERT should be used for all maintenance of the DIGTCERT, DIGTRING, and DIGTNMAP class profiles.

The RACDCERT command is a RACF TSO command used to:

- List information about the certificates for a specified RACF-defined user ID, or your own user ID.
- Add a certificate definition and associate it with a specified RACF-defined user ID, or your own user ID, and set the TRUST flag.
- Alter the TRUST flag or the LABEL name for a definition.
- Delete a definition.
- List a certificate contained in a data set and determine if it is associated with a RACF-defined user ID.
- Add or remove a certificate from a key ring.
- Create, delete, or list a key ring.
- Generate a public/private key pair and certificate.
- Write a certificate to a data set.
- Create a certificate request.
- Create, alter, delete, or list a user ID mapping.

Issuing Options

The following table identifies the eligible options for issuing the RACDCERT command:

Table 37. How the RACDCERT Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	No	See Rules	See Rules	No

Rules:

- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT.*target-node*.DIGTCERT.APPL and AUTODIRECT.*target-node*.DIGTRING.APPL, where *target-node* is the remote node to which the update is to be propagated.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

Authorization Required

To issue the RACDCERT command, you must have one of the following authorities:

- SPECIAL

- Sufficient authority to resource IRR.DIGTCERT.*function* in the FACILITY class, as described in Table 38, Table 41 on page 272, Table 46 on page 277, Table 47 on page 279, and Table 48 on page 279.

Authority required for the RACDCERT functions: The authorities required to perform the various RACDCERT functions are summarized in Table 38. Except for the CHECKCERT, EXPORT and GENCERT functions, the authorities listed are to the resource IRR.DIGTCERT.*function* in the FACILITY class. CHECKCERT looks at IRR.DIGTCERT.LIST, EXPORT looks at IRR.DIGTCERT.EXPORT or IRR.DIGTCERT.EXPORTKEY, and GENCERT looks at both IRR.DIGTCERT.ADD and IRR.DIGTCERT.GENCERT.

Table 38. RACDCERT Authority Checks

FUNCTION	READ	UPDATE	CONTROL
ADD	Add a certificate to one's own user ID.	Add a certificate for someone else.	Add a certificate-authority or site certificate.
ADDRING	Create a key ring for one's own user ID.	Create a key ring for another user ID.	Not applicable.
ALTER	Change the trust status or label of one's own certificate.	Change the trust status or label of someone else's certificate.	Change the trust status or label of a certificate authority or site certificate.
ALTMAP	Alter a mapping associated with one's own user ID	Alter a mapping associated with another user ID or MULTIID.	Not applicable.
CHECKCERT	Check one's own certificate.	Check the certificate of someone else.	Check the certificate-authority or site certificates.
CONNECT	See Table 47 on page 279.	See Table 47 on page 279.	See Table 48 on page 279.
DELETE	Delete one's own certificate.	Delete the certificate of someone else.	Delete a certificate-authority or site certificate.
DELMAP	Delete a mapping associated with one's own user ID	Delete a mapping associated with another user ID or MULTIID.	Not applicable.
DELRING	Delete one's own key ring.	Delete the key ring of someone else.	Not applicable.
EXPORT	See Table 46 on page 277.	See Table 46 on page 277.	See Table 46 on page 277.
GENCERT	See Table 41 on page 272.	See Table 41 on page 272.	See Table 41 on page 272.
GENREQ	Generate a request based on one's own certificate.	Generate a request based on the certificate of another user.	Generate a request based on a SITE or CERTAUTH certificate.
LIST	List one's own certificate.	List the certificate of someone else.	List certificate-authority or site certificates.
LISTMAP	List mapping information associated with one's own user ID	List mapping information associated with another user ID or MULTIID.	Not applicable.
LISTRING	See one's own key ring.	See the key ring of someone else.	Not applicable.
MAP	Create a mapping associated with one's own user ID.	Create a mapping associated with another user ID or MULTIID.	Not applicable.
REMOVE	Remove a certificate from one's own key ring.	Remove a certificate-authority or site certificate from one's own ring.	Remove a certificate from a ring of another.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the RACDCERT command is:

```
RACDCERT
[ ID(userid) | MULTIID | SITE | CERTAUTH ]
```

```

[ LIST
  [ (LABEL('label-name')) ]
  | [ (SERIALNUMBER(serial-number)
      [ ISSUERSDN('issuer's-dist-name')) ] ]
  | ADD(data-set-name)
    [ TRUST | NOTRUST | HIGHTRUST ]
    [ WITHLABEL('label-name') ]
    [ PASSWORD('pkcs12-password') ]
    [ ICSF ]
  | CHECKCERT(data-set-name)
    [ PASSWORD('pkcs12-password') ]
  | ALTER
    [ (LABEL('label-name')) ]
    | [ (SERIALNUMBER(serial-number)
        [ ISSUERSDN('issuer's-dist-name') ] ) ]
    [ TRUST | NOTRUST | HIGHTRUST ]
    [ NEWLABEL('label-name') ]
  | DELETE
    [ (LABEL('label-name')) ]
    | [ (SERIALNUMBER(serial-number)
        [ ISSUERSDN('issuer's-dist-name')) ] ]
  | GENCERT [ (request-data-set-name) ]
    [ SUBJECTSDN(
      [ CN('common-name') ]
      [ T('title') ]
      [ OU('organizational-unit-name1'
          [ , 'organizational-unit-name2',... ] ) ]
      [ O ('organization-name') ]
      [ L ('locality') ]
      [ SP ('state-or-province') ]
      [ C ('country') ] ) ]
    [ SIZE (key-size) ]
    [ NOTBEFORE (
      [ DATE (yyyy-mm-dd) ] [ TIME (hh:mm:ss) ] ) ]
    [ NOTAFTER (
      [ DATE(yyyy-mm-dd) ] [ TIME (hh:mm:ss) ] ) ]
    [ WITHLABEL ('label-name') ]
    [ SIGNWITH ( [ CERTAUTH | SITE ]
      LABEL ('label-name') ) ]
    [ PCICC | ICSF ]
    [ KEYUSAGE ( [ HANDSHAKE ] [ DATAENCRYPT ]
    [ DOCSIGN ] [ CERTSIGN ] ) ]
    [ ALTNAME ( IP(numeric-ip-address)
      DOMAIN('internet-domain-name')
      EMAIL('email-address')
      URI('universal-resource-identifier') ) ]
  | EXPORT (LABEL ('label-name') )
    DSN (output-data-set-name)
    [FORMAT (CERTDER | CERTB64
      | PKCS7DER | PKCS7B64
      | PKCS12DER | PKCS12B64) ]
    [ PASSWORD ('pkcs12-password') ]
  | GENREQ (LABEL('label-name'))
    DSN(output-data-set-name)
  | CONNECT( [ ID(userid) | SITE | CERTAUTH ]
    LABEL('label-name')
    RING(ring-name)
    [ DEFAULT ]
    [ USAGE(PERSONAL | SITE | CERTAUTH) ] )

```

```

| REMOVE( [ ID(userid) | SITE | CERTAUTH ]
          LABEL('label-name')
          RING(ring-name))
| ADDRING(ring-name)
| DELRING(ring-name)
| LISTRING[(ring-name)]
]
| MAP [ (data-set-name) ]
      [ SDNFILTER('subject's-distinguished-name-filter') ]
      [ IDNFILTER('issuer's-distinguished-name-filter') ]
      [ CRITERIA(criteria-profile-name-template) ]
      [ WITHLABEL('label-name') ]
      [ TRUST | NOTRUST ]
| ALTMAP [ (LABEL('label-name') ) ]
          [ NEWCRITERIA(criteria-profile-name-template) ]
          [ NEWLABEL('label-name') ]
          [ TRUST | NOTRUST ]
| DELMAP [ (LABEL('label-name') ) ]
| LISTMAP [ (LABEL('label-name') ) ]
| [DEBUG]

```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

Parameters

On the RACDCERT command, you can specify the ID, MULTIID, SITE, or CERTAUTH keywords to identify the user ID that will be associated with the function being performed by the RACDCERT command. ID, CERTAUTH, and SITE can be specified with the keywords LIST, ADD, DELETE, GENCERT, GENREQ, and EXPORT. The ID and MULTIID keywords can be specified with the keywords MAP, ALTMAP, DELMAP, and LISTMAP. The ID keyword can also be specified with the keywords ADDRING, DELRING, LISTRING, CONNECT, and REMOVE. The function CHECKCERT ignores the keywords ID, MULTIID, CERTAUTH, and SITE.

If more than one function keyword is specified, the last one specified is processed first, followed by the others. Extraneous keywords that are not related to the function being performed are ignored.

If the DIGTCERT, DIGTNMAP, DIGTRING, or DIGTCRIT class is RACLISTed, whenever you perform a RACDCERT ADD, ALTER, DELETE, PERMIT, REMOVE, MAP, ALTMAP, DELMAP, ADDRING, or DELRING, you should refresh the class with the RACF TSO command:

```
SETROPTS RACLIST(DIGTCERT, DIGTNMAP) REFRESH
```

ID(userid) |
MULTIID |
CERTAUTH |
SITE

Specifies the user ID associated with the digital certificate, key ring, or certificate mapping. If more than one of these keywords is specified, the last one specified will be processed, and the others will be ignored as part of TSO command parsing. If none of these keywords are specified, the ID keyword with the user ID of the user issuing the command is the default.

RACDCERT

For certificate management you can specify a specific user ID's certificate, a certificate-authority certificate, or a site certificate through the ID(*userid*), CERTAUTH, or SITE keyword respectively. For ring management, you can only specify a user ID associated with the key ring because only user IDs can have key rings.

For mapping operations (MAP, ALTMAP, DELMAP, and LISTMAP), these keywords indicate the type of mapping that is being processed. When ID is specified, the mapping operation is associated with the specified user ID. When MULTIID is specified, the user ID is determined by additional criteria and the CRITERIA keyword must be specified. The SITE and CERTAUTH keywords are not permitted for the mapping operands.

LIST(SERIALNUMBER(*serial-number*) ISSUERSDN(*'issuer's-dist-name'*))
LIST(LABEL(*'label-name'*))

Displays the digital certificate information, including certificate authority and site certificate information. For each digital certificate defined, the following information is displayed:

- Label
- Certificate ID
- Status (trusted, not trusted, or highly trusted)
- Validity dates
- Serial number
- Issuer's distinguished name
- Up to 256 bytes of the subject's name, as found in the certificate itself
- Extensions, if present (specifically, keyUsage and subjectAltName)
- Type of private key (ICSF, non-ICSF or PCICC), or NONE if there is no private key
- Private key size
- Ring Associations, if present (the ring name to which this certificate is connected and the ring owner)

If the RACDCERT command is issued with no other operands, it lists the command issuer's digital certificate information. If the RACDCERT command is issued with the ID keyword and no other operands, it lists the digital certificate information associated with the user ID specified with the ID keyword.

The issuer's distinguished name and the subject's distinguished name can contain blanks. If the name displayed in the output is subsequently entered with the ISSUERSDN keyword, the blanks must be included. In the output of LIST, the characters '>' and '<' are used as delimiters to mark the beginning and end of the serial number, issuer's name and subject's name. When information continues to the next line, '<' appears in column 79 of the output, and '>' appears in column 9 of the continuation line.

If the user has only one certificate, or if all certificates are to be displayed, the SERIALNUMBER and ISSUERSDN keywords, or the LABEL keyword, and their associated values can be omitted. If the user has more than one certificate the LABEL, SERIALNUMBER, or SERIALNUMBER and ISSUERSDN can be used to select which certificate to list.

Note: The ISSUERSDN and LABEL keywords are case and blank sensitive, therefore, the values specified for these keywords need to be exact when being specified.

For a description of *label-name*, see the description of the WITHLABEL keyword for the ADD keyword.

If present, the SubjectAltName values are displayed under the heading **Subject's AltNames**. The subheadings **IP**, **EMail**, **Domain**, and **URI** are followed by the appropriate values. If more than one line is required to display the value, the additional lines will start in the same column. The word "at" replaces the @ symbol for *email-address*. For example,

EMail: JRoenick at US.Mycompany.Com-More-Info-About-An-EMail-Address-follows-Some-More-Info-About-An-EMail-Address

If present, the KeyUsage values are displayed next to the heading **Key Usage**. The possible values are:

- HANDSHAKE, which indicates digitalSignature and keyEncipherment are on
- DATAENCRYPT, which indicates dataEncipherment is on
- DOCSIGN, which indicates nonRepudiation is on
- CERTSIGN, which indicates keyCertSign and cRLSign is on

The KeyUsage values are displayed as GENCERT keywords separated by commas. For example:

Key Usage: HANDSHAKE, CERTSIGN

Note: Original RACF digital certificate support did not allow for labels to be specified. These certificates display "No label assigned" in the label field when listed.

ADD(*data-set-name*)

Specifies that a digital certificate is to be defined. The specified data set must contain the digital certificate. The data set containing the digital certificate or certificate package must be cataloged, and cannot be a PDS or a PDS member. The RECFM expected by RACDCERT is VB. When the ADD keyword is specified, RACDCERT dynamically allocates and opens the specified data set, and reads the certificate from it as binary data.

Note: The issuer of the RACDCERT command must have READ access to the *data-set-name* data set to prevent an authorization abend from occurring when the data set is read.

Each user ID can be associated with more than one digital certificate but they must be added individually. The specified data set should contain only one digital certificate. The command reads the certificate from the data set, updates the user's profile, and creates the DIGTCERT profile.

If the certificate being added already exists in the RACF database, RACF will accept the certificate (and refresh the stored information) provided all of the following conditions are true:

1. The certificate is being added to the same User ID, SITE, or CERTAUTH as before.
2. The label specified for the certificate matches the old value or no label is specified.

Otherwise, an informational message is issued, and the certificate is not re-added.

RACDCERT

The ADD keyword also supports certificate replacement, for cases of certificate renewal and fulfillment by an external certificate authority. When a certificate is replaced, all existing information is updated to reflect the new certificate. This includes key ring connections.

The certificate in the RACF database is replaced if the following conditions are true.

- If the existing certificate has a private key associated with it:
 - The certificate is being added to the same User ID, SITE, or CERTAUTH as before
 - The certificate being added is not a duplicate (i.e., not a simple re-add)
 - The certificate being added is not expired.
 - The certificate being added has the same public key as the existing certificate.
- If the existing certificate does not have a private key associated with it:
 - The certificate is being added to the same User ID, SITE, or CERTAUTH as before
 - The certificate being added is not a duplicate.
 - The certificate being added is not expired.
 - The certificate being added has a later expiration date than that of the existing certificate
 - The certificate being added has the same subject's distinguished name, issuer's distinguished name, and public key as the existing certificate.

Note: When a certificate is being replaced, a new label can be specified.

Supported certificate package formats

The certificate package must be in one of the following formats:

1. A single BER encoded X.509 certificate.
2. A single Base64 encoded X.509 certificate.
3. A Privacy Enhanced Mail (PEM) encoded X.509 certificate. If the input is in this format, only the Originator Certificate is used.
4. One or more X.509 certificates contained within a PKCS#7 DER encoding package.
5. One or more X.509 certificates and private keys contained within a PKCS#12 DER encoding package. If the input is in this format, only the first user certificate and private key is used. PKCS#12 is also known as Private Information Exchange (PFX). The obsolete PFX V0.02 standard is not supported.

Details regarding all certificates

The following are additional details regarding RACDCERT's certificate processing:

1. All fields as defined for X.509 version 1 certificates must be present and must have a length greater than zero (non-null).
2. X.509 certificates with version numbers greater than 3 are not supported.
3. Noncritical extensions are ignored. Critical extensions that are supported include:
 - keyUsage - { 2 5 29 15}
 - basicConstraints - { 2 5 29 19 }

- subjectAltname - { 2 5 29 17 }
 - issuerAltName - { 2 5 29 18 }
 - certificatePolicies - { 2 5 29 32 }
 - policyMappings - { 2 5 29 33 }
 - policyConstraints - { 2 5 29 36 }
 - nameConstraints - { 2 5 29 30 }
 - extKeyUsage - { 2 5 29 37 }
 - hostIdMapping - { 1 3 18 0 2 18 1 }
 - subjectKeyIdentifier - { 2 5 29 14 }
 - authorityKeyIdentifier - { 2 5 29 35 }
4. Subject and issuer names can contain only the following string types:
 - UTF8 - TAG 12 (7-bit ASCII only)
 - PRINTABLESTRING - TAG 19
 - T61STRING - TAG 20
 - IA5STRING - TAG 22
 - VISIBLESTRING - TAG 26
 - GENERALSTRING - TAG 27
 - BMPString - TAG 30 (ASCII Unicode only)
 5. Because certificates can be encoded differently, be aware when transporting the different certificate encodings to and from z/OS and OS/390 systems. Both the BER encoded X.509 and PKCS#7 formats are binary formats and must be transported in their exact binary format. For example, binary formats, such as BER and X.509, cannot have any ASCII to EBCDIC translation performed on them, therefore, they must be transported in their exact binary format. In contrast, text formats, such as PEM and Base64, must be transported as text. When transporting for an ASCII system, the ASCII to EBCDIC translation must be performed for the PEM format and Base64 format certificate.

PKCS#7 format details

The ADD function of RACDCERT can accept a PKCS#7 certificate package. If there is more than one certificate in the package, the set consisting of the 2nd through last certificate is the hierarchy of CAs. If the command issuer is authorized to add CERTAUTH certificates, CA certificates will be added to the CERTAUTH category in the hierarchy order (top CA down to lowest CA). Thus each certificate in the package may be verified using its previously added parent. The LABEL for each added CA certificate will be generated, however the LABEL value specified will not be used. Any certificate that does not represent a link in the chain is discarded. If the command issuer is not authorized to add CERTAUTH certificates, an informational message will be issued. In either case, processing will then continue with the first certificate in the package (the end-entity certificate). For each CERTAUTH certificate in the PKCS#7 package, the following list of rules will apply to determine the trust status. The list is in priority order. For rules that conflict, the first matching rule wins:

Rules:

1. If the certificate is already defined to RACF with status HIGHTRUST, the certificate retains its HIGHTRUST status.
2. The trust status specified by the command issuer will apply to the top CA certificate. This primes the chain with a trust value which may be

RACDCERT

inherited down. (See the next rule.) The HIGHTRUST keyword is ignored if CERTAUTH is not specified.

3. For all lower CA certificates in the PKCS#7 package and for the top CA certificate when no trust status is specified, the trust status is determined dynamically as follows:
 - a. If NOTRUST is specified by the command issuer, the certificate is added with NOTRUST status.
 - b. If the certificate has one or more of the following inconsistencies, the certificate is added with NOTRUST status:
 - 1) The certificate is expired.
 - 2) The certificate has an incorrect date range relative to the issuing CA certificate. (The validity period is not completely contained with the validity period of the issuing CA certificate.)
 - 3) The issuer of the certificate is missing from the PKCS#7 package and is not already installed under CERTAUTH.
 - 4) The certificate has an unknown signature algorithm.
 - c. If no inconsistencies are detected, the certificate is added and inherits the trust status of its parent. If the certificate's parent has not previously been added (either as a part of this package or otherwise), the certificate is added with TRUST status if it is self-signed, NOTRUST status if it is not self-signed.
4. HIGHTRUST will be inherited from the parent as per the previous rule only if CERTAUTH and HIGHTRUST are both specified. In all other cases HIGHTRUST reverts to TRUST when inheriting from the parent.

The authority required to add the CERTAUTH certificates from a PKCS#7 is the same authority required to add CERTAUTH certificates directly, either CONTROL authority to IRR.DIGTCERT.ADD in the FACILITY class or RACF SPECIAL.

Note: There is no backout support for PKCS#7 add error processing. If a terminating error is encountered during processing, any CERTAUTH certificate previously added is not removed. Unless otherwise stated in the error message description, any error messages issued are relative to the certificate where the error occurred. This may be the end-entity certificate or one of the CERTAUTH certificates.

PKCS#12 format details

PKCS#12 certificate packages can be processed. These certificates are encrypted with a password when received and must be decrypted with the same password before being added to the RACF database. (See keyword 268.) A PKCS#12 certificate package can contain more than one certificate. RACDCERT ADD only processes the first certificate in the PKCS#12 certificate package that has a private key.

When adding a certificate package that contains a private key, and ICSF is being used to store private keys, ADD creates an ICSF key label in the format `IRR.DIGTCERT.userid.cvtsname.ebcdic-stck-value`, where *userid* is the owning user ID, *cvtsname* is the system name, as taken from the CVT, and *ebcdic-stck-value* is an EBCDIC version of the current store clock value.

TRUST | NOTRUST | HIGHTRUST

When specified with the ADD keyword, indicates whether the status of the certificate being added is trusted, not trusted, or highly trusted. Whether a

certificate is not trusted or trusted depends on whether or not the certificate is valid and whether the private key has been compromised or not.

Since highly trusted certificates are by definition trusted certificates, any certificate usage that was enabled by marking the certificate trusted will also be enabled by marking the certificate highly trusted. However, only certificate-authority certificates can be highly trusted. The trust status is stored in the UACC field of the certificate profile:

- X'00' indicates the status is trusted
- X'80' indicates the status is not trusted
- X'C0' indicates the status is highly trusted

When a certificate is trusted, it can be used by RACF for its intended purpose (map to a user ID, or treat as a trusted certificate authority or trusted site).

For a personal certificate, TRUST indicates that the certificate can be used to authenticate a user ID.

For a certificate-authority certificate, a trusted certificate is one that can be used to authenticate a user's certificate by indicating that the entity identified in the certificate (for example, the certificate authority) can issue certificates that this system honors. This implies that a user can gain access to the system based on the information contained in the certificate if the user's certificate was signed by a trusted certificate authority.

For site certificates, a trusted certificate is one indicating that the entity identified in the certificate (for example, the site) can gain access to the system based on information contained within the certificate. Since the authority that issued the certificate might not be defined to the system as a certificate authority, this certificate information might not be able to be authenticated.

TRUST should only be specified if the command issuer knows:

- This is a valid certificate for this user, site, or certificate authority.
- The private key related to this certificate has not been compromised.

If no trust value is specified on the command, the following processing will take place to determine the trust status:

- If the certificate's signature can be verified, the certificate has not expired, and the certificate's validity date range is within the validity date range of the certifying authority's certificate, the trust status is set to the trust status of the certifying authority's certificate. For self-signed certificates the certificate being added is set to TRUST by default.
- If the certificate has expired, has an incorrect validity date range, or cannot be verified because it either has an unknown encryption algorithm or RACF cannot locate its certifying authority's certificate, the status is set to NOTRUST by default.
- If the trust status is to be set from the status of the certifying certificate and the certifying certificate is highly trusted, the status will be trusted.

If the certificate's signature is incorrect, the certificate is not added.

This keyword is unrelated to the trusted attribute as defined in the started procedures table (ICHRIN03).

RACDCERT

WITHLABEL(*'label-name'*)

specifies the label to be associated with the certificate. Up to 32 characters can be specified. The *label-name* can contain blanks and mixed-case characters.

This label is used as a “handle” instead of the serial number and issuer’s distinguished name. It can be used to store a descriptive text.

If the value specified in WITHLABEL already exists, RACDCERT returns a message indicating that the label has already been used. The certificate is not added.

If the user did not specify WITHLABEL, and the data set being processed is PKCS#12, the label is extracted from the PKCS#12 package and truncated to 32 characters if required.

If WITHLABEL is not specified, or extracted from the PKCS#12 package, RACDCERT generates a label for the certificate. The generated label is of the form LABELnnnnnnnn, where “nnnnnnnn” is the first integer value, starting at “00000001” that generates a unique label name.

The *label-name* is stripped of leading and trailing blanks. If a single quotation mark is intended to be part of the *label-name*, you must use two single quotation marks together for each single quotation mark within the string, and the entire string must then be enclosed within single quotation marks.

PASSWORD(*'pkcs12-password'*)

specifies the password that is associated with the PKCS#12 certificate package. This keyword is required if the data set is PKCS#12 and it must not be specified if the data set is not PKCS#12.

Note: The password specified will be visible on the screen, so care should be taken to prevent it from being viewed when entered. Because PKCS#12 passwords do not follow the normal TSO/E rules for password content, they cannot be suppressed as they normally would be.

The *'pkcs12-password'* can be up to 255 characters in length, is case sensitive, and can contain blanks.

ICSF

specifies that RACF should attempt to store the private key associated with this certificate in the ICSF PKDS. This includes when the key is introduced to RACF by issuing the ADD keyword for PKCS#12 certificate packages, and when an existing certificate profile containing a non-ICSF private key is replaced by issuing the ADD keyword.

This keyword is ignored if no private key is involved or if the private key already exists in the PKDS as an ICSF or PCICC key. If the ICSF keyword is not specified, the key is stored in the RACF database as a non-ICSF key. If the ICSF keyword is specified and ICSF is not operational or is not configured for PKA operations, processing stops and an error message is displayed. If the key is stored the ICSF PKDS, RACF stores a label (which refers to the key) in the RACF database. Any system using the key in the future would be required to have ICSF operational and configured for PKA operations with this PKDS.

CHECKCERT(*data-set-name*)

specifies that a digital certificate, contained in the data set *data-set-name* is to be evaluated to see if it has already been added to the RACF database, and associated with a user ID.

CHECKCERT lists the certificate in the specified data set. If the certificate request is made by a user with proper authority, information in the RACF database pertaining to that certificate is also displayed. Additionally, an authority check is performed by data management when the data set is opened.

The CHECKCERT keyword also supports the evaluation of site certificates and certificate authority certificates. It indicates if the certificate is defined and to whom it is defined after checking the resource IRR.DIGTCERT.LIST in the FACILITY class. READ authority is required if the certificate is associated with the user issuing the command. UPDATE authority is required if the certificate is associated with a user other than the issuer of the command. CONTROL authority is required if the certificate is a certificate authority or a site certificate.

The CHECKCERT keyword can be used on the same set of certificate packages that is allowed by RACDCERT ADD. See the ADD keyword on the RACDCERT command for more information.

CHECKCERT ignores the ID() parameter.

Notes:

1. The issuer of the RACDCERT command must have READ access to the *data-set-name* data set to prevent an authorization abend from occurring when the data set is read.
2. No certificate ID is displayed if the certificate is not installed. If the certificate is installed, the certificate ID is displayed only if the certificate has a label and the user is authorized to list the specific certificate information.

PASSWORD('pks12-password')

specifies the password that is associated with the PKCS#12 certificate package. It is required if the data set contains a PKCS#12 certificate package and it must not be specified if the data set contents are not PKCS#12.

Note: The password specified will be visible on the screen, so care should be taken to prevent it from being viewed when entered. Because PKCS#12 passwords do not follow the normal TSO/E rules for password content, they cannot be suppressed as they normally would be.

The 'pks12-password' can be up to 255 characters in length, is case sensitive, and can contain blanks.

ALTER(SERIALNUMBER(*serial-number*) ISSUERSDN('issuer's-dist-name')) ALTER(LABEL('label-name'))

Specifies that the status or the label of a digital certificate is to be changed for the specified user ID, certificate-authority certificate, or site certificate. The TRUST, NOTRUST, or NEWLABEL keyword must be specified with the ALTER keyword. If the user has only one certificate, the SERIALNUMBER and ISSUERSDN keywords, or the LABEL keyword, and their associated values can be omitted. If the user has more than one certificate the LABEL, SERIALNUMBER, or SERIALNUMBER and ISSUERSDN must be used to select which certificate to alter. When specifying the issuer's distinguished name

RACDCERT

or the label, the case and blanks displayed when the digital certificate information is listed must be maintained in the ISSUERSDN or the LABEL keyword.

For a description of *label-name*, see WITHLABEL(*'label-name'*) subkeyword for ADD(*data-set-name*).

Note that the only alterable certificate information is the TRUST status or the label of a certificate.

TRUST | NOTRUST | HIGHTRUST

When specified with the ALTER keyword, indicates whether the status of the certificate being altered is trusted, not trusted or highly trusted. If TRUST, NOTRUST, or HIGHTRUST is not specified with the ALTER keyword, no change to the status of the certificate is attempted.

For a detailed description, see TRUST | NOTRUST | HIGHTRUST subkeyword for ADD(*data-set-name*).

NEWLABEL(*'new-label-name'*)

specifies the label replacing the previous label (if there was one specified) that is assigned to a certificate. See WITHLABEL(*'label-name'*) subkeyword on the ADD(*data-set-name*) keyword for information on label rules.

If *new-label-name* is the same as *label-name*, the label is not changed and no message is issued.

DELETE(SERIALNUMBER(*serial-number*) ISSUERSDN(*'issuer's-dist-name'*)) DELETE(LABEL(*'label-name'*))

Specifies that the digital certificate is to be deleted for the specified user ID, certificate-authority certificate, or site certificate. If the user has only one certificate, the SERIALNUMBER and ISSUERSDN keywords, or the LABEL keyword, and their associated values can be omitted. If the user has more than one certificate the LABEL, SERIALNUMBER, or SERIALNUMBER and ISSUERSDN must be used to select which certificate to delete. When specifying the issuer's distinguished name or the label, the mixed-case characters and blanks displayed when the digital certificate information is listed must be maintained in the ISSUERSDN or the LABEL keyword.

The DELETE keyword also supports site and certificate-authority certificates, and the deletion of the private key and other certificate data that is stored when the certificate was created.

For a description of *label-name*, see the description of the WITHLABEL subkeyword for the ADD keyword.

When a user profile is deleted with the DELUSER command, related DIGTCERT, DIGTRING, and DIGTNMAP profiles are deleted as a part of DELUSER processing. However, under some circumstances, residual profiles might not be deleted. For example, if you issue the DELUSER command from a down-level system (Version 2 Release 7 or earlier) that does not fully support the current level of digital certificate information, the profiles might not be deleted. The DELETE, DELRING and DELMAP keywords for RACDCERT support the specification of a user ID in order to allow residual certificate information related to the user ID to be deleted. The other RACDCERT functions, however, require the user ID to be defined to RACF.

GENCERT(*request-data-set-name*)

creates a digital certificate and potentially a public or private key pair. *Request-data-set-name* is the name of an optional data set that contains the PKCS#10 certificate request data. The request data contains the user's

generated public key and X.509 distinguished name. The request data must be signed, DER-encoded, and then Base64 encoded according to the PKCS#10 standard.

The subkeywords of the GENCERT keyword specify the information that is to be contained within the certificate that is being created.

If *request-data-set-name* is specified, RACDCERT does not generate a key pair because this data set contains the user's public key.

Request-data-set-name has characteristics (for example, RECFM) identical to the data set that can be specified with the ADD and CHECKCERT keywords. If *request-data-set-name* is specified, SIGNWITH must also be specified because the *request-data-set-name* data set does not contain a private key. If SIGNWITH is not specified, an informational message is issued. Note that the issuer of the RACDCERT command must have READ access to the *request-data-set-name* data set to prevent an authorization abend from occurring when the data set is read.

If *request-data-set-name* is specified and extensions are present and not overridden by other keywords specified with the RACDCERT command, they are copied to the certificate being created. These extensions and the logic involved with using them can be found:

- For subjectKeyIdentifier, see Table 39.
- For authorityKeyIdentifier, see Table 40.
- For keyUsage, see Table 42 on page 276.
- For basicConstraints, see Table 43 on page 276.
- For subjectAltName, see Table 44 on page 277.
- For issuerAltName, see Table 45 on page 277.

Table 39. subjectKeyIdentifier Extension Logic for GENCERT

When the Requested Data Set is Specified	When the Requested Data Set is not specified
The extension is encoded using the subjectKeyIdentifier value from the requested data set if present, if not present the extension is encoded by generating the keyIdentifier according to the Public Key Infrastructure Standards.	The extension is encoded by generating the keyIdentifier according to Public Key Infrastructure Standards.

Table 40. authorityKeyIdentifier Extension Logic for GENCERT

When SIGNWITH is specified	When SIGNWITH is not specified
The extension is encoded using the subjectKeyIdentifier value of the signing certificate if present, if not present the extension is not created.	The authorityKeyIdentifier extension is not created.

Authority Required for the GENCERT Function: The GENCERT keyword allows a certificate to be generated and signed. Effective controls on the user ID that is being associated with the certificate and what certificate is being used to sign the generated certificate are essential.

RACF performs two checks that determine the authority required for the GENCERT command:

1. How the certificate is being signed, specified with the SIGNWITH keyword.

RACDCERT

Users with SPECIAL authority can use the SIGNWITH keyword with any value. Users without SPECIAL authority must have authority to the IRR.DIGTCERT.GENCERT resource in the FACILITY class. If SIGNWITH is specified without the CERTAUTH or SITE keyword, the certificate is signed with the certificate identified with the LABEL keyword for the user who is issuing the RACDCERT command. This requires READ access to the resource IRR.DIGTCERT.GENCERT in the FACILITY class. If either SIGNWITH(CERTAUTH...) or SIGNWITH(SITE) is specified, CONTROL authority is required to the resource IRR.DIGTCERT.GENCERT in the FACILITY class.

Not specifying SIGNWITH indicates that the certificate is to be self-signed. The signing key is owned by the certificate itself. Thus the authority needed for signing is determined by what type of certificate is being generated. Generating a self-signed certificate for one's self requires READ access to the resource IRR.DIGTCERT.GENCERT in the FACILITY class. Generating a self-signed certificate for another user requires UPDATE access to the resource IRR.DIGTCERT.GENCERT in the FACILITY class. Generating a self-signed certificate for either SITE or CERTAUTH requires CONTROL access to IRR.DIGTCERT.GENCERT in the FACILITY class.

2. What type of certificate is being generated, which is specified with the ID(), SITE or CERTAUTH keywords.

Users with SPECIAL authority can generate a digital certificate for any RACF-defined user or for any certificate-authority or site certificate. Users without SPECIAL authority can generate certificate authority or site certificates if they have CONTROL authority to the resource IRR.DIGTCERT.ADD in the FACILITY class. Users without SPECIAL authority can generate certificates for other users if they have UPDATE authority to the resource IRR.DIGTCERT.ADD in the FACILITY class. Users without SPECIAL authority can generate certificates for themselves if they have READ authority to the resource IRR.DIGTCERT.ADD in the FACILITY class.

Table 41. Authority Required To Generate a Certificate

SIGNWITH	Own Certificate	Someone Else's Certificate	SITE or CERTAUTH Certificate
SIGNWITH one's own certificate	READ authority to IRR.DIGTCERT.ADD and READ authority to IRR.DIGTCERT.GENCERT	UPDATE authority to IRR.DIGTCERT.ADD and READ authority to IRR.DIGTCERT.GENCERT	CONTROL authority to IRR.DIGTCERT.ADD and READ authority to IRR.DIGTCERT.GENCERT
SIGNWITH a SITE or CERTAUTH certificate	READ authority to IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.GENCERT	UPDATE authority to IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.GENCERT	CONTROL authority to IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.GENCERT
SIGNWITH not specified	READ authority to IRR.DIGTCERT.ADD and READ authority to IRR.DIGTCERT.GENCERT	UPDATE authority to IRR.DIGTCERT.ADD and UPDATE authority to IRR.DIGTCERT.GENCERT	CONTROL authority to IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.GENCERT

SUBJECTSDN

specifies the subject's X.509 distinguished name, which consists of the following components:

Common Name	Specified with the CN subkeyword.
Title	Specified with the T subkeyword.

Organizational Unit	Specified with the OU subkeyword. Multiple values can be specified for the organizational unit.
Organization	Specified with the O subkeyword.
Locality	Specified with the L subkeyword.
State/Province	Specified with the SP subkeyword.
Country	Specified with the C subkeyword.

SUBJECTSDN completely overrides the values contained in the certificate request in the data set specified with the GENCERT keyword.

Each of the elements in SUBJECTSDN is limited to 64 characters. Each of the SUBJECTSDN subkeywords can be specified only once. If the certificate being created is a self-signed certificate, the total length of the subject's distinguished name must be 229 characters or less. For non-self-signed certificates, the length of the subject's distinguished name is limited to 255 characters. These lengths include the X.509 identifiers (such as C= and CN=) and the dot qualifiers.

If the SUBJECTSDN name is too long, an informational message is issued and the certificate is not added.

Any printable character that can be mapped to an ASCII character can be specified. Characters that cannot be mapped, such as X'4A' (cent sign) and X'00' are shown by RACDCERT LIST as blanks.

If SUBJECTSDN and *request-data-set-name* are not specified, the programmer name data from the ID() user (either specified or defaulted), or the programmer name from the SITE or CERTAUTH anchor user IDs (irrsitec or irrcerta) is used as the common name (CN). If the programmer name is all blanks (X'40'), nulls (X'00'), pound signs (X'7B'), or X'FF' characters, the common name is set to the user ID that is to be associated with this certificate.

SIZE(*key-size*)

specifies the size of the private key expressed in decimal bits. Valid values range from 512 to 999 999, with the default being 1024. RACF does not enforce a limit on the SIZE value other than the 512 lower and 999 999 upper limits. If the SIZE specified is too large to be handled by RACF's key generation code, an informational message is issued and processing stops. The maximum key size is determined by United States export regulations and is controlled by non-RACF code in z/OS and OS/390.

Currently, the standard sizes of a key are:

Size	Description
512	Low-strength key.
768	Medium-strength key.
1024	High-strength key.

If the GENCERT keyword creates a public/private key pair and ICSF is being used to store private keys, GENCERT creates an ICSF key label in the format `IRR.DIGTCERT.userid.cvtsname.ebcdic-stck-value`, where *userid* is the owning user ID, *cvtsname* is the system name as taken from the CVT,

RACDCERT

and *ebcdic-stck-value* is an EBCDIC version of the current store clock value. If the key is associated with a certificate-authority certificate, *userid* is set to CERTIFAUTH. If the key is associated with a site certificate, then *userid* is set to SITECERTIF. Access to ICSF keys is controlled by profiles in the CSFKEYS and CSFSERV classes. See *z/OS ICSF Administrator's Guide* for more information.

NOTBEFORE(DATE(*yyyy-mm-dd*) TIME(*hh:mm:ss*))

specifies the local date and time from which the certificate is valid. If DATE(*yyyy-mm-dd*) is not specified, it defaults to the current local date. If TIME(*hh:mm:ss*) is not specified, it defaults to TIME (00:00:00). Note that *yyyy* must be in the range 1950 to 2040.

The time and date values are stored in the certificate as a universal time coordinated (UTC) value. The calculated UTC value can be incorrect if the NOTBEFORE/NOTAFTER date and time values represent a time that has a different local offset from UTC.

Note that the use of the date format *yyyy-mm-dd* is valid. However, to aid installations familiar with the RACF date format, the value can be specified in the format *yyyy/mm/dd*.

NOTAFTER(DATE(*yyyy-mm-dd*) TIME(*hh:mm:ss*))

specifies the local date and time after which the certificate is no longer valid. If DATE(*yyyy-mm-dd*) is not specified it defaults to one year from the NOTBEFORE(DATE()) value. If TIME(*hh:mm:ss*) is not specified, it defaults to TIME(23:59:59). Note that *yyyy* must be in the range 1950 to 2040 if specified, and 1951 to 2041 if defaulted.

The time and date values are stored in the certificate as a universal time coordinated (UTC) value. The calculated UTC value can be incorrect if the NOTBEFORE/NOTAFTER date and time value represent a time that has a different local offset from UTC.

The NOTBEFORE value must be earlier than the NOTAFTER value or an informational message is issued.

Note the use of the date format *yyyy-mm-dd* is valid. However, to aid installations familiar with the RACF date format, the value can be specified as *yyyy/mm/dd*.

WITHLABEL('label-name')

specifies the label assigned to this certificate. If specified, this must be unique to the user ID with which the certificate is associated. If not specified, it defaults in the same manner as the WITHLABEL keyword on the RACDCERT ADD command.

The *label-name* is stripped of leading and trailing blanks. If a single quotation mark is intended to be part of the *label-name*, you must use two single quotation marks together for each single quotation mark within the string, and the entire string must then be enclosed within single quotation marks.

See the WITHLABEL subkeyword on the ADD keyword for information on label rules.

SIGNWITH(CERTAUTH LABEL('label-name'))

SIGNWITH(SITE LABEL('label-name'))

SIGNWITH(LABEL('label-name'))

specifies the certificate with a private key that is signing the certificate. If not specified, the default is to sign the certificate with the private key of the

certificate that is being generated. This creates a self-signed certificate. The signing certificate must belong to the userid executing the command.

If SIGNWITH is specified, it must refer to a certificate that has a private key associated with it. If no private key is associated with the certificate, an informational message is issued and processing stops. If

request-data-set-name is specified on the GENCERT keyword, the SIGNWITH keyword is also required.

Note that self-signed certificates are always trusted, while all other certificates are created with the trust status of the certificate specified in the SIGNWITH keyword. If the certificate specified in the SIGNWITH keyword is not trusted, an informational message is issued but the certificate is still generated.

PCICC | ICSF

Specifies how RACF should generate the key pair and how the private key should be stored for future use. If GENCERT is issued without a *request-data-set-name*, a key pair is to be generated. If PCICC is specified, the key pair is generated using the PCI cryptographic coprocessor. If ICSF is specified, the key pair is generated using software. In either case the resulting private key is stored in the ICSF PKDS.

If neither keyword is specified, the key pair is generated using software and the private key is stored in the RACF database as a non-ICSF key.

If either keyword is specified and ICSF is not operational or is not configured for PKA operations, processing stops and an error message is displayed. If the PCICC keyword is specified, a PCI cryptographic coprocessor must also be present and operational. Otherwise processing stops and an error message is displayed.

If the key is stored as either an ICSF key or a PCICC key, any system using the key in the future would be required to have ICSF operational and configured for PKA operations with this PKDS. For a PCICC key, any system using the key in the future would also need to have a PCI cryptographic coprocessor present and operational with this PKDS.

KEYUSAGE

specifies the appropriate values for the KeyUsage certificate extension, of which one or more of the values might be coded. For certificate authority certificates, the default is CERTSIGN and is **always** set. There is no default for certificates that are not certificate-authority certificates.

HANDSHAKE

facilitates identification and key exchange during security handshakes, such as SSL, which set the digitalSignature and keyEncipherment indicators.

DATAENCRYPT

encrypts data, which sets the dataEncipherment indicator.

DOCSIGN

specifies a legally binding signature, which sets the nonRepudiation indicator.

CERTSIGN

specifies a signature for other digital certificates and CRLs, which sets the keyCertSign and cRLSign indicators.

RACDCERT

Table 42. *keyUsage Extension Logic for GENCERT*

Situation	keyUsage Present in Requested Data Set	keyUsage not Present in Requested Data Set
When KEYUSAGE is specified and the target ID is CERTAUTH	If the certSign bit is turned off in the request dataset, the request will fail. Otherwise the extension is encoded as requested by the RACDCERT invoker. Additionally, the certSign and cRLSign bits will be turned on if not already specified by the CERTSIGN keyword.	The extension is encoded as requested by the RACDCERT invoker. Additionally, the certSign and cRLSign bits are turned on.
When KEYUSAGE is specified and the target ID is SITE or ID(<i>userid</i>)	The extension is encoded as requested by the RACDCERT invoker.	The extension is encoded as requested by the RACDCERT invoker.
When KEYUSAGE is not specified and the target ID is CERTAUTH	If the certSign bit is turned off this command fails, otherwise the extension is encoded as specified in the requested dataset.	The extension is encoded by turning the certSign and cRLSign bits on.
When KEYUSAGE is not specified and the target ID is SITE or ID(<i>userid</i>)	The extension is encoded using the requested data set values.	The keyUsage extension is not created.

Table 43. *basicConstraints Extension Logic for GENCERT*

Situation	basicConstraints Present in Requested Data Set	basicConstraints not Present in Requested Data Set
When the target ID is CERTAUTH	If the cA boolean value is false, the command will fail. Otherwise the extension is encoded turning the cA bit on. Pathlength is not included.	The extension is encoded turning the cA bit on. Pathlength is not included.
When the target ID is SITE or ID(<i>userid</i>)	The extension is encoded using the requested data set values, including the pathLength.	The basicConstraints extension is not created.

ALTNAME

specifies the appropriate values for the subjectAltName extension, of which one or more of the values might be coded. If required for the extension, RACF converts the entered values to ASCII.

Note: RACF assumes the terminal codepage is IBM-1047 and translates to ASCII accordingly.

IP(*'numeric-ip-address'*)

specifies a quoted string containing a fully qualified *'numeric-ip-address'* in IPV4 dotted decimal form, which is four decimal numbers (each number must be a value from 0-255) separated by periods:

9.117.2.45

DOMAIN(*'internet-domain-name'*)

specifies a quoted string containing a fully qualified *'internet-domain-name'* (such as *'www.widgits.com'*). RACF does not check this value's validity.

EMAIL(*'email-address'*)

specifies a quoted string containing a fully qualified *'email-address'*, such as *'jasper at moes.bar.com'*. RACF replaces the word "at" with the @ symbol (x'7C') to conform with RFC822. If RACF cannot locate the word "at" it assumes the address is already in RFC822 form and makes no attempt to alter it other than converting it to ASCII.

URI('universal-resource-identifier')

specifies the 'universal-resource-identifier' (such as 'http://www.widgits.com'). RACF does not check the validity of this value.

Table 44. subjectAltName Extension Logic for GENCERT

Situation	subjectAltName Present in Requested Data Set	subjectAltName not Present in Requested Data Set
When ALTNAME is specified	The extension is encoded as requested by the RACDCERT invoker.	The extension is encoded as requested by the RACDCERT invoker.
When ALTNAME is not specified	The extension is encoded using the requested data set values.	The subjectAltName extension is not created.

Table 45. issuerAltName Extension Logic for GENCERT

When SIGNWITH is specified	When SIGNWITH is not specified
The extension is encoded using the subjectAltName value of the signing certificate if the extension is present. Otherwise, the issuerAltName extension is not created.	The IssuerAltName extension is not created.

EXPORT(LABEL('label-name'))

Writes a certificate to a data set. *Label-name* identifies the certificate that is being exported. Depending on which keyword you specify, you can export a certificate, a certificate and its CA chain, or a certificate and private key.

Table 46. Authority Required to Export a Certificate Package

Export in CERT format	Export one's own certificate: READ authority to IRR.DIGTCERT.EXPORT	Export the certificate of another user: UPDATE authority to IRR.DIGTCERT.EXPORT	Export SITE or CERTAUTH's certificate: CONTROL authority to IRR.DIGTCERT.EXPORT
Export in PKCS7 format	Export own's certificate, but not the parent CA chain: READ authority to IRR.DIGTCERT.EXPORT	Export other's certificate, but not the parent CA chain: UPDATE authority to IRR.DIGTCERT.EXPORT	Export SITE or CERTAUTH's certificate and/or the entire parent CA chain: CONTROL authority to IRR.DIGTCERT.EXPORT
Export in PKCS12 format	Export own's certificate and the private key: READ authority to IRR.DIGTCERT.EXPORTKEY	Export other's certificate and the private key: CONTROL authority to IRR.DIGTCERT.EXPORTKEY	Export SITE or CERTAUTH's certificate and the private key: CONTROL authority to IRR.DIGTCERT.EXPORTKEY

DSN(output-data-set-name)

specifies the data set that is to contain the certificate. The data set *output-data-set-name* is deleted and reallocated if it exists. If EXPORT is specified, DSN must also be specified.

FORMAT(CERTB64 | CERTDER | PKCS7B64 | PKCS7DER | PKCS12B64 | PKCS12DER)

Specifies the format of the exported certificate package. Valid values for FORMAT are:

CERTB64

specifies a DER encoded X.509 certificate that has been encoded using Base64.

CERTDER

specifies a DER encoded X.509 certificate.

PKCS7B64

specifies a DER encoded PKCS#7 package that has been encoded using Base64.

RACDCERT

PKCS7DER

specifies a DER encoded PKCS#7 package.

PKCS12B64

specifies a DER encoded PKCS#12 package that has been encoded using Base64.

PKCS12DER

specifies a DER encoded PKCS#12 package.

Note: PKCS12B64 is the default if PASSWORD is specified; otherwise, CERTB64 is the default.

The CERT keywords indicate that only a certificate is to be exported.

The PKCS7 keywords indicate to export a certificate and its CA chain. If the command issuer is authorized to export CERTAUTH certificates PKCS#7 processing will attempt to package any certificate authority certificate necessary to complete the basing chain to the exported certificate. If a certificate in the chain cannot be found under CERTAUTH or is expired or the command issuer is not authorize to export CERTAUTH certificates, an informational message will be issued. Processing continues creating an incomplete PKCS#7 package. An incomplete PKCS#7 package can still be processed by RACF but may or may not be useful for OEM products.

The PKCS#12 keywords indicate to export the certificate and the private key (which must exist and must not be an ICSF or PCICC key). The package produced by specifying one of the PKCS12 keywords is encrypted using the password specified according to the PKCS#12 standard.

PKCS#12 processing will attempt to package any certificate-authority certificate necessary to complete the basing chain to the exported certificate. If a certificate in the chain cannot be found under CERTAUTH, an informational message will be issued. Processing continues and an incomplete PKCS#12 package is created that can still be processed by RACF but may or may not be useful for OEM products.

PASSWORD('pkcs12-password')

Specifies the password to use for PKCS#12 package encryption. The string is converted before being used, so any characters entered must be translatable to 7-bit ASCII. However, RACF does not enforce this.

Note: RACF assumes the current host codepage is 1047 and translates to ASCII accordingly.

GENREQ(LABEL('label-name'))

creates a PKCS#10 Base64-encoded certificate request and writes it to a data set. This request contains the subject's distinguished name and public key, and is signed with the private key associated with the specified certificate. Additionally, the following extensions are copied to the certificate request if they are present in the certificate:

- subjectAltName
- subjectKeyIdentifier
- authorityKeyIdentifier
- basicConstraints
- keyUsage

Typically, these requests are sent to a certificate authority; however, they can also be imported into and signed by RACF using the GENCERT keyword with a *request-data-set-name*.

GENREQ requires that the certificate have a private key associated with it. If no private key is associated with the certificate, an informational message is issued and processing stops. *Label-name* identifies the certificate.

DSN(*output-data-set-name*)

specifies the data set that is to contain the certificate request. The data set *output-data-set-name* is deleted and reallocated if it exists. If you specify GENREQ, DSN must also be specified.

CONNECT(ID(*userid*) LABEL('label-name') RING(*ring-name*))

CONNECT(SITE LABEL('label-name') RING(*ring-name*))

CONNECT(CERTAUTH LABEL('label-name') RING(*ring-name*))

specifies that a digital certificate is being added to a key ring. This certificate must be added to the RACF database by a RACDCERT ADD or RACDCERT GENCERT command prior to issuing the CONNECT command. ID(*userid*) indicates that the certificate being added to the key ring is a user certificate, and *userid* is the user ID that is associated with this certificate. If the ID keyword is not specified, it defaults to the value specified or the default value on the RACDCERT command. SITE indicates that the certificate being added to the key ring is a site certificate. CERTAUTH indicates that the certificate being added to the key ring is a certificate-authority certificate.

Authority Required for the CONNECT Function: The USAGE keyword allows a certificate to be connected to a ring and used in a manner that differs from the certificate's original use. For example, a certificate that is a user certificate could be used as a certificate-authority certificate.

The USAGE keyword is powerful, and must be controlled. The rules for connection are shown in Table 47, which shows the access control checks that are performed when connecting to one's own key ring, and Table 48, which shows the access control checks that are performed when connecting to someone else's key ring.

Table 47. Authority Required to Connect to One's Own Key Ring

USAGE	Own Certificate	Someone Else's Certificate	SITE or CERTAUTH Certificate
PERSONAL	READ authority to IRR.DIGTCERT.CONNECT	UPDATE authority to IRR.DIGTCERT.CONNECT	UPDATE authority to IRR.DIGTCERT.CONNECT
SITE/CERTAUTH	CONTROL authority to IRR.DIGTCERT.ADD and READ authority to IRR.DIGTCERT.CONNECT	CONTROL authority to IRR.DIGTCERT.ADD and UPDATE authority to IRR.DIGTCERT.CONNECT	UPDATE authority to IRR.DIGTCERT.CONNECT

Table 48. Authority Required To Connect to Someone Else's Key Ring

USAGE	Own Certificate	Someone Else's Certificate	SITE or CERTAUTH Certificate
PERSONAL	CONTROL authority to IRR.DIGTCERT.CONNECT	CONTROL authority to IRR.DIGTCERT.CONNECT	CONTROL authority to IRR.DIGTCERT.CONNECT
SITE/CERTAUTH	CONTROL authority to IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.CONNECT	CONTROL authority to IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.CONNECT	CONTROL authority to IRR.DIGTCERT.CONNECT

RACDCERT

See the USAGE subkeyword below for additional information on the authority required to change a certificate's usage.

LABEL('label-name')

specifies the certificate that is being added to the key ring. When specifying the CONNECT keyword, LABEL must also be specified.

RING(ring-name)

specifies the ring to which this certificate is being added. When specifying the CONNECT keyword, RING must also be specified.

Note: The ring belongs to the ID associated with the RACDCERT command. The certificate being connected belongs to the ID indicated in the CONNECT subkeyword.

DEFAULT

specifies that the certificate is the default certificate for the ring. Only one certificate within the key ring can be the default certificate. If a default certificate already exists, its DEFAULT status is removed, and the specified certificate becomes the default certificate. If you want the specified certificate to be the default, DEFAULT must be explicitly specified.

If you have a key ring with a default certificate and you want to remove the default status of the certificate without defining another certificate as the default certificate, CONNECT the certificate again without specifying the DEFAULT keyword.

USAGE(PERSONAL)

USAGE(SITE)

USAGE(CERTAUTH)

specifies how this certificate is used within the specified ring. If no usage is specified, the usage is the same as the certificate that is being connected.

The USAGE keyword allows the altering of the trust policy within the confines of a specific key ring. For example, a CERTAUTH certificate connected with USAGE(PERSONAL) can be used to demote a certificate-authority certificate in order to insure that it is not used as a certificate authority in this ring. It can be used as a user certificate if a private key is present. However, typically one would not be present. Consequently, connecting a CERTAUTH certificate as USAGE(PERSONAL) is a way of marking it NOTRUST for this key ring only. Also, a user certificate connected with USAGE(CERTAUTH) can be used to promote an ordinary user certificate to a certificate-authority certificate. It can then be used to authenticate user certificates for this key ring only.

For the sake of consistency, other certificate and USAGE variations are supported. However, there is currently no practical application for them.

When using the USAGE keyword to change the usage of a certificate, such as is done when a PERSONAL certificate is being used as a SITE or CERTAUTH certificate, RACDCERT must ensure that you have the ability to define a SITE or CERTAUTH certificate by authenticating that the command issuer has CONTROL authority to the resource IRR.DIGTCERT.ADD in the FACILITY class. This ensures that a user cannot bypass the installation security policy through the use of USAGE.

REMOVE(ID(userid) LABEL('label-name') RING(ring-name))

REMOVE(SITE LABEL('label-name') RING(ring-name))

REMOVE(CERTAUTH LABEL('label-name') RING(ring-name))

specifies that a digital certificate is being removed from a key ring. ID(userid)

indicates that the certificate being removed is a user certificate, and *userid* is the user ID that is associated with this certificate. If the ID keyword is not specified, it defaults to the value that is specified or defaulted to on the RACDCERT command. SITE indicates that this is a site certificate, and CERTAUTH indicates that this is a certificate authority certificate.

LABEL(*'label-name'*)

identifies the certificate that is being removed from the key ring. When specifying the REMOVE keyword, LABEL must also be specified.

RING(*ring-name*)

identifies the ring from which this certificate is being removed. When specifying the REMOVE keyword, RING must also be specified.

Note: The ring belongs to the ID associated with the RACDCERT command, not the REMOVE subkeyword.

ADDRING(*ring-name*)

specifies the creation of a key ring. *Ring-name* is the name of the key ring being created. This key ring must not already exist for this user. Key ring names become names of RACF profiles in the DIGTRING class, and can contain only characters that are allowed in RACF profile names. Although asterisks are allowed in ring-names, a single asterisk is not allowed.

Lower case characters are permitted. A key ring name can be up to 237 characters in length. Since only user IDs can have key rings, neither CERTAUTH nor SITE can be specified with ADDRING.

DELRING(*ring-name*)

specifies the deletion of a key ring. *Ring-name* is the name of the key ring. Since only user IDs can have key rings, neither CERTAUTH nor SITE can be specified with DELRING.

Note that when a DELUSER command is issued against a user ID, all of the key rings that are owned by that user ID are also deleted.

LISTRING

LISTRING(*ring-name* / *)

specifies listing a key ring. *Ring-name* is the name of the key ring. To list all rings that are associated with a particular user, LISTRING (*) must be specified. For each certificate in the ring, the following information is displayed:

- The ring name,
- The owner of the certificate (ID(name), CERTAUTH, or SITE)
- The label assigned to the certificate,
- The DEFAULT status of the certificate within the ring, and
- The usage within the ring.

Since only user IDs can have key rings, neither CERTAUTH nor SITE can be specified with LISTRING.

MAP

MAP(*data-set-name*)

specifies that a certificate name filter is to be defined. It results in the creation of a profile in the DIGTNMAP class. DIGTNMAP profiles are used as filters when a user attempts to access the system using a digital certificate. A user ID is found by comparing the issuer's distinguished name and subject's distinguished name from the certificate with the filter values used to create the DIGTNMAP profile. The user ID is specified with the ID keyword or specified in DIGTCRIT profiles if MULTIID is specified. When you specify MAP, you must also specify IDNFILTER, SDNFILTER, or both.

RACDCERT

A data set name can be specified with the MAP keyword. *Data-set-name* is the name of the data set that contains a certificate. The certificate provides a model for the filter names specified with SDNFILTER and IDNFILTER. The subject's distinguished name is used beginning with the value specified by SDNFILTER. The issuer's distinguished name is used beginning with the value specified by IDNFILTER. Using a model certificate is optional but can reduce the chance of typographical errors when entering long filters for SDNFILTER or IDNFILTER.

The model certificate used with the MAP keyword can have an issuer's distinguished name or subject's distinguished name that exceeds 255 characters. However, the portion of each used in the filter to associate a user ID with the certificate cannot exceed 255 characters.

See the ADD keyword for acceptable certificate formats.

Data-set-name has the same characteristics (for example, RECFM) as the dataset that can be specified with the ADD and CHECKCERT keywords. The issuer of the RACDCERT command must have READ access to the data set containing the *data-set-name*.

IDNFILTER('issuer's-distinguished-name-filter')

specifies the significant portion of the issuer's distinguished name that is used as a filter when associating a user ID with a certificate. For an explanation of how filter values are used to associate a user ID with a digital certificate, see *z/OS Security Server RACF Security Administrator's Guide*.

When specified without *data-set-name* on the MAP keyword, you must specify the entire portion of the distinguished name to be used as a filter.

The format of the *issuer's-distinguished-name-filter* is similar to the output displayed when a certificate is listed with RACDCERT. It is an X.509 distinguished name in an address type format:

component.component.component.component...

Or, more specifically:

qualifier1=node1.qualifier2=node2....qualifiern=noden

For example:

IDNFILTER('OU=Class 1 Certificate.O=BobsCertAuth, Inc.L=internet.C=US')

The value specified for IDNFILTER must begin with a prefix found in the following list, followed by an equal sign (X'7E'). Each component should be separated by a period (X'4B'). The case, blanks, and punctuation displayed when the digital certificate information is listed must be maintained in the IDNFILTER. Since digital certificates only contain characters available in the ASCII character set, the same characters should be used for the IDNFILTER value. Valid prefixes are:

Country	Specified as C=
State/Province	Specified as SP=
Locality	Specified as L=
Organization	Specified as O=
Organizational Unit	Specified as OU=
Title	Specified as T=
Common Name	Specified as CN=

When specified along with *data-set-name* on the MAP keyword, the *issuer's-distinguished-name-filter* must correspond to a starting point within the issuer's distinguished name found in the certificate contained in the data set. You should specify enough of the name to precisely identify the starting point for the filter. For example, if the certificate in the data set has the following issuer:

```
OU=Class 1 Certificate.O=BobsCertAuth, Inc.L=internet.C=US
```

and you want all certificates issued by
BobsCertAuth

to be selected by this filter, you specify:

```
IDNFILTER('O=BobsCertAuth')
```

Without the data set containing the certificate, you need to enter the following to produce the same result:

```
IDNFILTER('O=BobsCertAuth, Inc.L=internet.C=US')
```

A maximum of 255 characters can be entered for IDNFILTER. When a starting point value is specified for a certificate contained in a data set, there cannot be more than 255 characters between the starting point and the end of the issuer's name in the certificate.

IDNFILTER is optional if SDNFILTER is specified. If IDNFILTER is not specified, only the subject's name is used as a filter. If IDNFILTER is specified and only a portion of the issuer's name is to be used as the filter, SDNFILTER must not be specified.

If both IDNFILTER and SDNFILTER are specified, the IDNFILTER value does not need to begin with a valid prefix from the list above. This allows the use of certificates from a certificate authority that chooses to include non-standard data in the issuer's distinguished name.

SDNFILTER('subject's-distinguished-name-filter')

specifies the significant portion of the subject's distinguished name. This is the part of the name that will be used as a filter when associating a user ID with a certificate. For an explanation of how filter values are used to associate a user ID with a digital certificate, see *z/OS Security Server RACF Security Administrator's Guide*.

When specified without *data-set-name* on the MAP keyword, you must specify the entire portion of the distinguished name to be used as the filter.

The format of the *subject's-distinguished-name-filter* is similar to the output displayed when a certificate is listed with RACDCERT. It is an X.509 distinguished name in an address type format:

```
component.component.component.component...
```

Or, more specifically:

```
qualifier1=node1.qualifier2=node2....qualifiern=noden
```

For example:

```
SDNFILTER('CN=Bob Smith.OU=BobsAccountingDept.O=BobsMart.L=internet')
```

The value specified for SDNFILTER must begin with a prefix found in the following list, followed by an equal sign (X'7E'). Each component should be

separated by a period (X'4B'). The case, blanks, and punctuation displayed when the digital certificate information is listed must be maintained in the SDNFILTER. Since digital certificates only contain characters available in the ASCII character set, the same characters should be used for the SDNFILTER value. Valid prefixes are:

Country	Specified as C=
State/Province	Specified as SP=
Locality	Specified as L=
Organization	Specified as O=
Organizational Unit	Specified as OU=
Title	Specified as T=
Common Name	Specified as CN=

When specified along with *data-set-name* on the MAP keyword, the *subject's-distinguished-name-filter* must correspond to a starting point within the subject's distinguished name found in the certificate contained in the data set. You should specify enough of the name to precisely identify the starting point for the filter. For example, if the certificate in the data set has the following subject:

```
CN=Bob Smith.OU=BobsAccountingDept.O=BobsMart.L=internet
```

and you want all certificates for anyone in
BobsAccountingDept

to be selected by this filter, you specify:

```
SDNFILTER('OU=BobsAcc')
```

Without the data set containing the certificate, you need to enter the following to produce the same result:

```
SDNFILTER('OU=BobsAccountingDept.O=BobsMart.L=internet')
```

A maximum of 255 characters can be entered for SDNFILTER. When a starting point value is specified for a certificate contained in a data set, there cannot be more than 255 characters between the starting point and ending point of the subject's name in the certificate.

SDNFILTER is optional if IDNFILTER is specified. If SDNFILTER is not specified, only the issuer's name is used as a filter. SDNFILTER must not be specified with IDNFILTER unless the value of IDNFILTER will result in the entire issuer's name being used in the filter. Note that subject's name can be partial but cannot be used in a filter that contains only a partial issuer's name.

CRITERIA(*criteria-profile-name-template*)

when specified with MULTIID, it indicates a dynamic user ID mapping. The user ID associated with this mapping profile is based not only on the issuer's distinguished name and the subject's distinguished name found in the certificate, but also on additional criteria. The *criteria-profile-name-template* specifies the additional criteria in the form of a profile name containing one or more variable names, separated by freeform text. These variable names begin with an ampersand (&) and end with a period. The freeform text should identify the variables contained in the template:

```
variable-name1=&variable-name1.variable-name2=&variable-name2...
```

For example, if the application identity and system identifier are to be considered in determining the user ID associated with this mapping, the CRITERIA keyword should be specified as follows:

```
CRITERIA(APPLID=&APPLID.SYSID=&SYSID)
```

The RACF-defined criteria are the application ID (APPLID) and the system-identifier (SYSID). When a user presents a certificate to the system for identification, the identity of the application (as well as the system the user is trying to access) being accessed becomes part of the criteria. The application passes its identity to RACF, and RACF determines the system-identifier. The system-identifier is the 4-character value specified for the SID parameter of the SMFPRMxx member of SYS1.PARMLIB. These values are substituted for &APPLID and &SYSID in the criteria.

Once the substitution is made, the fully expanded criteria template is used as a resource name to find a matching profile defined in the DIGTCRIT class using the RDEFINE command. For example, if the application being accessed is BANKU on system SYSA, the template is:

```
APPLID=BANKU.SYSID=SYSA
```

You should define a profile in the DIGTCRIT class using the RDEFINE command for this name. The user ID to be associated with these certificates must be specified as the APPLDATA. While the DIGTCRIT profile name can be discrete, generic profiles can be used if you have generic profile checking active for the DIGTCRIT class. A DIGTCRIT profile name of APPLID=BANKU.* allows the certificates to be used on any system, rather than just system SYSA. While generic characters such as *

and

%

can be used when defining the DIGTCRIT class profiles, they should not be used in the template name specified with the CRITERIA keyword.

Criteria names other than APPLID and SYSID are allowed, but are effective in Certificate Name Filtering if the application supplies these criteria names and their associated values to RACF when the user attempts to access the application using a certificate. SYSID is determined by RACF, but APPLID must be specified with the initACEE callable service. Criteria names, such as APPLID and SYSID, should only be specified on RACDCERT if the application instructs you to do so.

A maximum of 255 characters can be entered when specifying the CRITERIA keyword. The values can be entered in any case, but are made upper case by the RACDCERT command because they must match upper case profile names in the DIGTCRIT class to be effective. When specifying the criteria value, note that the maximum profile name length in the DIGTCRIT class is 246.

The CRITERIA keyword can only be set for MULTIID.

WITHLABEL('label-name')

specifies the label that is assigned to this mapping. If specified, it must be

RACDCERT

unique to the user ID with which the mapping is associated. If WITHLABEL is not specified, a label is generated in the same manner as issuing the WITHLABEL keyword for the RACDCERT ADD command.

Up to 32 characters can be specified for *label-name*. It can contain embedded blanks and mixed-case characters, and is stripped of leading and trailing blanks. If a single quotation mark is intended to be part of the *label-name*, you must use two single quotation marks together for each single quotation mark within the string, and the entire string must then be enclosed within single quotation marks.

TRUST | NOTRUST

when specified with MAP, indicates whether this mapping can be used to associate a user ID to a certificate presented by a user accessing the system. If neither TRUST nor NOTRUST is specified, the default is TRUST.

ALTMAP

ALTMAP(LABEL('label-name'))

changes the label, trust status, or criteria associated with the mapping identified by *label-name*. Specifying *label name* is required if more than one mapping is associated with the user ID. If NEWLABEL, NEWCRITERIA, or TRUST/NOTRUST is not specified, the mapping is not altered.

NEWCRITERIA(criteria-profile-name-template)

changes the template associated with this mapping when specified with MULTIID. New DIGTCRIT profiles must be created to match the new template profile names. NEWCRITERIA can only be set for MULTIID.

NEWLABEL('new-label-name')

specifies the label replacing the previous label assigned to a certificate mapping. See the WITHLABEL subkeyword on the ADD keyword for information on label rules.

If *new-label-name* is the same as *label-name*, the label is not changed and no message is issued.

TRUST | NOTRUST

when specified with ALTMAP, indicates whether this mapping can be used to associate a user ID to a certificate presented by a user accessing the system.

DELMAP

DELMAP(LABEL('label-name'))

deletes the mapping identified by *label-name* for the specified user ID. Specifying *label-name* is required if more than one mapping is associated with the user ID. Note that mappings might also be deleted as part of DELUSER processing.

LISTMAP

LISTMAP(LABEL('label-name'))

lists information about the mapping identified by *label-name* for the user ID specified. Do not specify LABEL if you intend to list all mappings associated with the user ID.

It is possible for LISTMAP to encounter an error in locating filter information in a DIGTNMAP profile. For example, if a previous RACDCERT command did not complete successfully due to a system failure, or early termination by the issuer, the user profile may still indicate that a filter exists, but the DIGTNMAP profile is not there. If the DIGTNMAP information is not found, the LISTMAP output will contain the text:

Filter with label *label-name* not found.

If this text is present in the LISTMAP output, a RACDCERT DELMAP specifying this label can be issued to remove the residual filter information from the user's profile.

DEBUG

displays additional diagnostic information pertaining to encryption calls and RACF invoked ICHEINTY ALTER, RACROUTE REQUEST=EXTRACT, and RACROUTE REQUEST=DEFINE failures. However, the contents of these messages will not be documented in any publication. When a problem is encountered, customers can use this keyword to gather diagnostic information for the IBM Support Center.

Controlling RACDCERT Examples: Effective use of RACDCERT requires that its privileges be carefully controlled. However, end users and application administrators should be allowed some flexibility in defining their security characteristics. These guidelines might prove useful.

- The ability to add certificate authorities should be allowed to only a small set of trusted people.
- End users should be permitted to add, delete, and modify the contents of their own key rings and add, delete, and alter their own certificates.
- Help desk personnel should be allowed the ability to list certificates and rings.

Assume that your system administrators, who are the only ones who are allowed to add, alter, or delete certificate-authority certificates or site certificates, are in the group WEBADMIN. Furthermore, assume that your help desk personnel are in the group HELPDESK. The commands in Figure 34 on page 288 show one method of controlling access to RACDCERT functions.

RACDCERT

RDEFINE	FACILITY	IRR.DIGTCERT.ADD	UACC(NONE)
RDEFINE	FACILITY	IRR.DIGTCERT.ADDRING	UACC(NONE)
RDEFINE	FACILITY	IRR.DIGTCERT.DELRING	UACC(NONE)
RDEFINE	FACILITY	IRR.DIGTCERT.LISTRING	UACC(NONE)
RDEFINE	FACILITY	IRR.DIGTCERT.CONNECT	UACC(NONE)
RDEFINE	FACILITY	IRR.DIGTCERT.REMOVE	UACC(NONE)
RDEFINE	FACILITY	IRR.DIGTCERT.LIST	UACC(NONE)
RDEFINE	FACILITY	IRR.DIGTCERT.ALTER	UACC(NONE)
RDEFINE	FACILITY	IRR.DIGTCERT.DELETE	UACC(NONE)
RDEFINE	FACILITY	IRR.DIGTCERT.GENCERT	UACC(NONE)
RDEFINE	FACILITY	IRR.DIGTCERT.GENREQ	UACC(NONE)
RDEFINE	FACILITY	IRR.DIGTCERT.EXPORT	UACC(NONE)
RDEFINE	FACILITY	IRR.DIGTCERT.EXPORTKEY	UACC(NONE)
RDEFINE	FACILITY	IRR.DIGTCERT.MAP	UACC(NONE)
RDEFINE	FACILITY	IRR.DIGTCERT.ALTMAP	UACC(NONE)
RDEFINE	FACILITY	IRR.DIGTCERT.DELMAP	UACC(NONE)
RDEFINE	FACILITY	IRR.DIGTCERT.LISTMAP	UACC(NONE)
PERMIT	IRR.DIGTCERT.ADDRING	CLASS(FACILITY)	ID(WEBADMIN) ACCESS(CONTROL)
PERMIT	IRR.DIGTCERT.DELRING	CLASS(FACILITY)	ID(WEBADMIN) ACCESS(CONTROL)
PERMIT	IRR.DIGTCERT.LISTRING	CLASS(FACILITY)	ID(WEBADMIN) ACCESS(CONTROL)
PERMIT	IRR.DIGTCERT.ADD	CLASS(FACILITY)	ID(WEBADMIN) ACCESS(CONTROL)
PERMIT	IRR.DIGTCERT.ALTER	CLASS(FACILITY)	ID(WEBADMIN) ACCESS(CONTROL)
PERMIT	IRR.DIGTCERT.DELETE	CLASS(FACILITY)	ID(WEBADMIN) ACCESS(CONTROL)
PERMIT	IRR.DIGTCERT.LIST	CLASS(FACILITY)	ID(WEBADMIN) ACCESS(CONTROL)
PERMIT	IRR.DIGTCERT.GENCERT	CLASS(FACILITY)	ID(WEBADMIN) ACCESS(CONTROL)
PERMIT	IRR.DIGTCERT.GENREQ	CLASS(FACILITY)	ID(WEBADMIN) ACCESS(CONTROL)
PERMIT	IRR.DIGTCERT.EXPORT	CLASS(FACILITY)	ID(WEBADMIN) ACCESS(CONTROL)
PERMIT	IRR.DIGTCERT.MAP	CLASS(FACILITY)	ID(WEBADMIN) ACCESS(CONTROL)
PERMIT	IRR.DIGTCERT.LISTMAP	CLASS(FACILITY)	ID(WEBADMIN) ACCESS(CONTROL)
PERMIT	IRR.DIGTCERT.ALTMAP	CLASS(FACILITY)	ID(WEBADMIN) ACCESS(CONTROL)
PERMIT	IRR.DIGTCERT.DELMAP	CLASS(FACILITY)	ID(WEBADMIN) ACCESS(CONTROL)
PERMIT	IRR.DIGTCERT.ADD	CLASS(FACILITY)	ID(*)
PERMIT	IRR.DIGTCERT.ALTER	CLASS(FACILITY)	ID(*)
PERMIT	IRR.DIGTCERT.DELETE	CLASS(FACILITY)	ID(*)
PERMIT	IRR.DIGTCERT.LIST	CLASS(FACILITY)	ID(*)
PERMIT	IRR.DIGTCERT.ADDRING	CLASS(FACILITY)	ID(*)
PERMIT	IRR.DIGTCERT.DELRING	CLASS(FACILITY)	ID(*)
PERMIT	IRR.DIGTCERT.LISTRING	CLASS(FACILITY)	ID(*)
PERMIT	IRR.DIGTCERT.CONNECT	CLASS(FACILITY)	ID(*)
PERMIT	IRR.DIGTCERT.REMOVE	CLASS(FACILITY)	ID(*)
PERMIT	IRR.DIGTCERT.MAP	CLASS(FACILITY)	ID(*)
PERMIT	IRR.DIGTCERT.LISTMAP	CLASS(FACILITY)	ID(*)
PERMIT	IRR.DIGTCERT.ALTMAP	CLASS(FACILITY)	ID(*)
PERMIT	IRR.DIGTCERT.DELMAP	CLASS(FACILITY)	ID(*)
PERMIT	IRR.DIGTCERT.LIST	CLASS(FACILITY)	ID(HELPDESK) ACCESS(CONTROL)
PERMIT	IRR.DIGTCERT.LISTRING	CLASS(FACILITY)	ID(HELPDESK) ACCESS(CONTROL)

Figure 34. Controlling Access to RACDCERT Functions

Examples

Table 49. RACDCERT Examples

Example 1	<p><i>Operation</i> User RACFADM with SPECIAL authority requests the addition of a digital certificate for user NETBOY. RACFADM has placed the digital certificate in data set 'RACFADM.NETBOY.CERT' and wants the status of the certificate to be trusted. RACFADM issues the following RACDCERT command:</p> <p><i>Known</i> User RACFADM has SPECIAL authority.</p>
	<p>RACFADM has placed the digital certificate in data set 'RACFADM.NETBOY.CERT'.</p> <p><i>Command</i> RACDCERT ID(NETBOY) ADD('RACFADM.NETBOY.CERT') TRUST WITHLABEL('Savings Account')</p> <p><i>Output</i> None.</p>
Example 2	<p><i>Operation</i> User NETBOY requests the listing of his Savings Account digital certificate to ensure it has been defined, and that it is marked trusted. He has READ authority to the FACILITY class profile IRR.DIGTCERT.LIST. He issues the RACDCERT command with the LIST keyword, specifying the label to identify his certificate:</p> <p><i>Known</i> User NETBOY has been given READ access to profile IRR.DIGTCERT.LIST in the FACILITY class.</p> <p><i>Command</i> RACDCERT LIST(LABEL('Savings Account'))</p> <p><i>Output</i> See Figure 35 on page 291.</p>
Example 3	<p><i>Operation</i> User NETADMN has a digital certificate in a data set, and is uncertain who it belongs to, and whether or not it has been defined. The digital certificate is in data set 'NETADMN.SOMEONZ.CERT'. NETADMN has UPDATE authority to the FACILITY class profile IRR.DIGTCERT.LIST. He issues the following RACDCERT, and the output he receives indicates that it has already been defined for user GTM:</p> <p><i>Known</i> User NETADMN has been given UPDATE access to profile IRR.DIGTCERT.LIST in the FACILITY class. NETADMN has placed the digital certificate in data set 'NETADMN.SOMEONZ.CERT'.</p> <p><i>Command</i> RACDCERT CHECKCERT('NETADMN.SOMEONZ.CERT')</p> <p><i>Output</i> None.</p>
Example 4	<p><i>Operation</i> User NETADMN has a digital certificate in a data set, and is uncertain who it belongs to, and whether or not it has been defined. The digital certificate is in data set 'NETADMN.SOMEELSZ.CERT'. NETADMN has UPDATE authority to the FACILITY class profile IRR.DIGTCERT.LIST. He issues the following RACDCERT, and the output he receives indicates that the certificate is not associated with a user ID:</p> <p><i>Known</i> User NETADMN has been given UPDATE access to profile IRR.DIGTCERT.LIST in the FACILITY class. NETADMN has placed the digital certificate in data set 'NETADMN.SOMEELSZ.CERT'.</p> <p><i>Command</i> RACDCERT CHECKCERT('NETADMN.SOMEELSZ.CERT')</p> <p><i>Output</i> See Figure 36 on page 292.</p>
Example 5	<p><i>Operation</i> User GEORGEM requests the listing of of all certificates associated with his user ID.</p> <p><i>Known</i> User ID GEORGEM has 3 certificates, one of which is not associated with any rings.</p> <p><i>Command</i> RACDCERT LIST</p> <p><i>Output</i> See Figure 37 on page 293.</p>
Example 6	<p><i>Operation</i> User GEORGEM requests the listing of his key rings.</p> <p><i>Known</i> User ID GEORGEM has three key rings with certificates and one key ring that has no certificates.</p> <p><i>Command</i> RACDCERT LISTRING(*)</p> <p><i>Output</i> See Figure 38 on page 294.</p>

RACDCERT

Table 49. RACDCERT Examples (continued)

Example 7	<i>Operation</i>	User RACFADM with SPECIAL authority requests the addition of a new mapping profile that will associate the user ID WEBUSER with all digital certificates issued by VeriSign for Class 1 Individual Subscribers. A certificate is not readily available in a data set.
	<i>Known</i>	User RACFADM has SPECIAL authority to the FACILITY class.
	<i>Command</i>	RACDCERT ID(WEBUSER) MAP IDNFILTER('OU=VeriSign Class 1 Individual Subscriber.O=VeriSign, Inc..L=Internet') WITHLABEL('Savings Account')
Example 8	<i>Output</i>	None.
	<i>Operation</i>	User RACFADM with SPECIAL authority requests the addition of a new mapping profile that will associate all members of department BWVA, who have VeriSign Class 1 Individual subscriber certificates, with the user ID BWVAUSR. All members of the department have the organizational unit BWVA (OU=BWVA) as the second node of the subject name in their certificates. A certificate belonging to one of the department member is available in the data set JJONES.DEPTCERT. The use of the certificates should not be allowed until the network administrator gives his approval, so this mapping is currently not trusted.
	<i>Known</i>	User RACFADM has SPECIAL authority to profile IRR.DIGTCERT.MAP in the FACILITY class.
Example 9	<i>Command</i>	RACDCERT ID(BWVAUSR) MAP('JJONES.DEPTCERT') IDNFILTER('OU=VeriSign Class 1') NOTRUST SDNFILTER('OU=BWVA') WITHLABEL('BWVA USERS')
	<i>Output</i>	None.
	<i>Operation</i>	User RACFADM with SPECIAL authority has been notified by the network administrator that the users in department BWVA can begin using their certificates to access the system. The mapping previously created with the label BWVA USERS can now be marked trusted.
Example 10	<i>Known</i>	User RACFADM has SPECIAL authority to the FACILITY class.
	<i>Command</i>	RACDCERT ID(BWVAUSR) ALTMAP(LABEL('BWVA USERS')) TRUST
	<i>Output</i>	None.
Example 11	<i>Operation</i>	User RACFADM with SPECIAL authority has been notified that departments BWVB and BWVA have merged. The members of BWVA will be issued new digital certificates.
	<i>Known</i>	User RACFADM has SPECIAL authority to the FACILITY class.
	<i>Command</i>	RACDCERT ID(BWVAUSR) DELMAP(LABEL('BWVA USERS'))
Example 11	<i>Output</i>	None.
	<i>Operation</i>	User CERTADM with ALTER authority to profile IRR.DIGTCERT.MAP in the FACILITY class has received a digital certificate and placed it in the data set 'CERTADM.MODEL.CERT'. BobsBank has contracted VeriSign, Inc. to create certificates like the one received. These certificates will be installed on the workstations of each bank teller, and used to access the banking application BANKAPP. All certificates must map to the user ID BANKU, which has access to the data sets containing the banking data. CERTADM uses this function to display the issuer's name and subject name from the certificate. The values are:
	<i>Known</i>	User CERTADM has ALTER authority to profiles IRR.DIGTCERT.MAP in the FACILITY class.
Example 11	<i>Command</i>	RDEF DIGTCRIT BOBS.APPLID1=BANKAPP APPLDATA('BANKU') RACDCERT MULTIID MAP(MODEL.CERT) IDNFILTER('OU=') SDNFILTER('CN=') CRITERIA(BOBS.APPLID1=&APPLID) WITHLABEL('Bobs Tellers')
	<i>Output</i>	None.

Table 49. RACDCERT Examples (continued)

Example 12	<i>Operation</i>	User RACFADM with SPECIAL authority to the profile IRR.DIGTCERT.LISTMAP would like to list the mapping information for user ID NET1ID.
	<i>Known</i>	NET1ID has one mapping associated with it.
	<i>Command</i>	RACDCERT ID(NET1ID) LISTMAP
Example 13	<i>Output</i>	See Figure 39 on page 294.
	<i>Operation</i>	User RACFADM with SPECIAL authority to the profile IRR.DIGTCERT.LISTMAP would like to list the mapping information for MULTIID.
	<i>Known</i>	MULTIID has several mappings associated with it, but only the one with this label name will be listed.
Example 14	<i>Command</i>	RACDCERT MULTIID LISTMAP(LABEL('NewAPPL ID Mapping'))
	<i>Output</i>	See Figure 40 on page 294.
	<i>Operation</i>	User RACFADM with SPECIAL authority requests the creation of a certificate-authority certificate, with values for the subjectAltName extension and the keyUsage extension, for the local certificate authority.
Example 15	<i>Known</i>	User RACFADM has SPECIAL authority to the profile 'IRR.DIGTCERT.*' in the FACILITY class.
	<i>Command</i>	RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('Local CA')) ALTNAME(IP(9.117.170.150) DOMAIN('www.widgits.com')) EMAIL('localca@www.widgits.com')) URI('http://www.widgits.com/welcome.html')) KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN CERTSIGN) WITHLABEL('Local PKIX CA')
	<i>Output</i>	None.
Example 16	<i>Operation</i>	User CADUDE with CONTROL access to FACILITY Class profile 'IRR.DIGTCERT.*' wishes to mark the local certificate authority highly trusted.
	<i>Known</i>	User CADUDE has CONTROL authority to the profile 'IRR.DIGTCERT.*' in the FACILITY class.
	<i>Command</i>	RACDCERT CERTAUTH ALTER(LABEL('Local PKIX CA')) HIGHTRUST
Example 17	<i>Output</i>	None.
	<i>Operation</i>	User CADUDE wants to list the information from the local certificate-authority certificate in Example 15.
	<i>Known</i>	User CADUDE has CONTROL authority to the profile 'IRR.DIGTCERT.*' in the FACILITY class.
Example 18	<i>Command</i>	RACDCERT CERTAUTH LIST(LABEL('Local PKIX CA'))
	<i>Output</i>	See Figure 41 on page 295.

```
RACDCERT LIST(LABEL('Savings Account'))
```

Digital certificate information for user NETB0Y:

```
Label: Savings Account
Certificate ID: 2QbVxePC1ujigaWJlYeiQMGDg5aklaNA
Status: TRUST
Serial Number:
>5D666C20207A6638727A413872D8413B<
Issuer's Name:
>OU=BobsBank Savers.O=BobsBank.L=Internet<
Subject's Name:
>CN=S.S.Smith.OU=Digital ID Class 1 - NetScape.OU=BobsBank Class 1 - S<
>avingsAcct.O=BobsBank.L=Internet<
```

Figure 35. Example 2: Output for the RACDCERT LIST Command

RACDCERT

```
RACDCERT CHECKCERT('NETADMN.SOMELSZ.CERT')
```

```
Serial Number:  
  >79<  
Issuer's Name:  
  >CN=BobsBank Class 2<  
Subject's Name:  
  >brchMGR@BobsBank.com.CN=J. Miles.T=Manager.OU=Branch2.O=BobsBank,INC.<  
  >..SP=NY.L=Internet.C=USA<
```

Figure 36. Example 4: Output for the RACDCERT CHECKCERT Command

Digital certificate information for user GEORGEM:

```

Label: New Cert Type - Ser # 00
Certificate ID: 2QfHxdbZx8XU1YWmQMOFmaNA46iXhUBgQOKFmUB7QPDw
Status: TRUST
Start Date: 1996/04/18 03:01:13
End Date: 1998/02/13 03:01:13
Serial Number:
>00<
Issuer's Name:
>OU=Internet Demo CertAuth.0=TheCert Software Inc.<
Subject's Name:
>OU=Internet Demo CertAuth.0=TheCert Software Inc.<
Private Key Type: ICSF
Private Key Size: 1024
Ring Associations:
Ring Owner: GEORGEM
Ring:
>GEORGEMsNewRing01<
Ring Owner: GEORGEM
Ring:
>GEORGEMsRing<

Label: New Type Cert - VsignC1
Certificate ID: 2QfHxdbZx8XU1YWmQ00o14VAw4WZo0BgQOWiiYeVw/FA
Status: TRUST
Start Date: 1998/04/22 23:23:26
End Date: 2001/01/15 23:23:26
Serial Number:
>3511A552906FE7D029A44019D411FC3E<
Issuer's Name:
>OU=Class 1 Public Primary Certification Authority.0=VeriSign, Inc..C=<
>US<
Subject's Name:
>OU=VeriSign Class 1 CertAuth - Individual Subscriber.0=VeriSign, Inc..L=Int<
>ernet<
Private Key Type: Non-ICSF
Private Key Size: 512
Ring Associations:
Ring Owner: GEORGEM
Ring:
>GEORGEMsNewRing01<

Label: New Type Cert - VsignC2
Certificate ID: 2QfHxdbZx8XU1YWmQ00o14VAw4WZo0BgQOWiiYeVw/JA
Status: NOTRUST
Start Date: 1998/03/19 15:39:52
End Date: 1999/03/19 15:39:52
Serial Number:
>50D35294912F79D315E32B31AC8548F0<
Issuer's Name:
>OU=Class 2 Public Primary Certification Authority.0=VeriSign, Inc..C=<
>US<
Subject's Name:
>OU=VeriSign Class 2 CertAuth - Individual Subscriber.0=VeriSign, Inc..L=Int<
>ernet<
Private Key Type: None
Ring Associations:
*** No rings associated ***

```

Figure 37. Example 5: Output from the LIST Command

RACDCERT

Digital ring information for user GEORGEM:

```
Ring:
>GEORGEMsNewRing01<
Certificate Label Name      Cert Owner      USAGE      DEFAULT
-----
New Cert Type - Ser # 00   ID(GEORGEM)     PERSONAL    YES
New Type Cert - VsignC1   ID(GEORGEM)     CERTAUTH    NO
New Type Cert - VsignC2   ID(GEORGEM)     SITE        NO
65                          ID(JOHNPN)      PERSONAL    NO

Ring:
>GEORGEMsRing<
Certificate Label Name      Cert Owner      USAGE      DEFAULT
-----
GEORGEM's Cert # 48       ID(GEORGEM)     PERSONAL    NO
GEORGEM's Cert # 84       ID(GEORGEM)     PERSONAL    NO
New Cert Type - Ser # 00   ID(GEORGEM)     PERSONAL    YES

Ring:
>GEORGEMsRing#2<
Certificate Label Name      Cert Owner      USAGE      DEFAULT
-----
GEORGEM's Cert # 84       ID(GEORGEM)     PERSONAL    NO
GEORGEM's Cert # 48       ID(GEORGEM)     PERSONAL    NO

Ring:
>GEORGEMsRing#3<
*** No certificates connected ***
```

Figure 38. Example 6: Output from the LISTRING Command

Mapping information for user NETIID:

```
Label: General Internet ID Map
Status: TRUST
Issuer's Name Filter:
>OU=Internet Demo CertAuth.0=BobsMart Software Inc.L=Internet<
Subject's Name Filter:
>L=Internet<
```

Figure 39. Example 12: Output from the LISTMAP Command

Mapping information for MULTIID:

```
Label: NewAPPL ID Mapping
Status: TRUST
Issuer's Name Filter:
>OU=Class 1 Public Primary Certification Authority.0=VeriSign, Inc..C=<
>US<
Subject's Name Filter:
><
Criteria:
APPLID=&APPLID
```

Figure 40. Example 13: Output from the LISTMAP LABEL Command

Digital certificate information for CERTAUTH:

```

Label: Local PKIX CA
Certificate ID: Sc9bjZwKwLNxKw2myumPlGy8iGzJQSYi/u35j0eyFe213XgGBMTsUvCW
Status: HIGHTRUST
Start Date: 1999/08/05 00:00:00
End Date: 2000/08/05 23:59:59
Serial Number:
    >00<
Issuer's Name:
    >CN=Local CA<
Subject's Name:
    >CN=Local CA<
Subject's AltNames:
IP: 9.117.170.150
EMail: localca at www.widgits.com
Domain: www.widgits.com
URI: http://www.wigits.com/welcome.html
Key Usage: HANDSHAKE, DATAENCRYPT, DOCSIGN, CERTSIGN
Private Key Type: Non-ICSF
Private Key Size: 1024
Ring Associations:
*** No rings associated *** >

```

Figure 41. Example 16: Output from the LIST Command

RACLINK (Administer User ID Associations)

Purpose

Use the RACLINK command to:

- Define, approve, and delete (undefine) an established or pending user ID association
- List information related to a user ID association
- Establish password synchronization between user IDs

Notes:

1. When the RACLINK command is issued from ISPF, the TSO command buffer (including password data) is written to the ISPLOG data set. As a result, you should not issue this command from ISPF or you must control the ISPLOG data set carefully.
2. If the RACLINK command is issued as a RACF operator command, the command and the password data is written to the system log. Therefore, use of RACLINK as a RACF operator command should either be controlled or you should issue the command as a TSO command.

Issuing Options

The following table identifies the eligible options for issuing the RACLINK command:

Table 50. How the RACLINK Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	No	No	No

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To add a user profile, see 48.
- To display information from a user profile, see “LISTUSER (List User Profile)” on page 223.
- To change a user profile, see “ALTUSER (Alter User Profile)” on page 115.
- To delete a user profile, see “DELUSER (Delete User Profile)” on page 189.
- To obtain a list of user profiles, see “SEARCH (Search RACF Database)” on page 408.

Authorization Required

You have the authority to issue the RACLINK command for your own user ID.

To issue the RACLINK DEFINE command you must also have sufficient authority to the proper profiles in the RRSFDATA class. For RACLINK DEFINE, this is the first security check performed. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

You can issue the RACLINK command for a user ID other than your own if you have the following authority over the user ID specified on the ID keyword:

- SPECIAL
- The profile is within the scope of a group in which you have the group-SPECIAL attribute
- profile owner

When the DEFINE keyword is specified and the command issuer has sufficient authority to perform the RACLINK command for the user ID, the user ID association is implicitly approved if:

- A valid password is supplied for the user ID specified on the DEFINE keyword.
- The command issuer has one of the following authorities over the user ID specified on the DEFINE keyword:
 - SPECIAL
 - The profile is within the scope of a group in which the command issuer has the group-SPECIAL attribute
 - The command issuer is the owner of the profile.
- The command issuer has an association with a user ID on the node specified on the DEFINE keyword. That association must be either a PEER association or a MANAGED association with the command issuer as the manager. The user ID with which the command issuer has the association must have one of the following authorities over the user ID specified on the DEFINE keyword:
 - SPECIAL
 - It is within the scope of a group that has the group-SPECIAL attribute
 - It is the owner of the profile

When issuing the RACLINK command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the RACLINK command is:

```
[subsystem-prefix]RACLINK
    [ ID(userid1...) ]
    [ LIST [ ( [ node_ ] . [ userid2* ] ...)
      | DEFINE( [ node ] .userid2 [ /password ] ....)
      | [MANAGED | PEER [ (NOPWSYNC | PWSYNC) ]
      | UNDEFINE( [ node ] .userid2...)
      | APPROVE( [ node ] .userid2...) ]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

RACLINK

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

ID(*userid1*...)

specifies the user for whom the RACLINK operation is to be performed. Specify one or more user IDs on the RRSF node from which the command is issued.

If this operand is not specified, the command defaults to the user issuing the command at the node where the command is issued.

LIST([*node].[*userid2**]...)**

specifies that a list of associations for *node.userid2* is to be displayed. If multiple user IDs are specified, then multiple lists are displayed, one for each user ID specified.

RACLINK LIST (*.*) is the default. RACLINK LIST (*.*) lists all user ID associations for the specified user ID or the issuer's user ID if the ID keyword is not specified.

If the node name is not specified, the default is the local node.

The node names you specify must have been defined as RRSF nodes with the TARGET command.

The following information is displayed for each user ID association:

- User ID association type
 - Peer association
 - Managed association (including whether the specified user ID is the managed user ID or the managing user ID)
- Password synchronization status
 - YES (password synchronization is active)
 - NO (password synchronization is inactive)
 - N/A (password synchronization is not applicable to a managed association)
- User ID association status
 - PENDING APPROVAL BY *userid* (waiting for *userid* to approve or reject the user ID association)
 - ESTABLISHED (the user ID association has been approved)
 - SYSTEM ERROR (an unexpected error occurred on the target node that prevented the user ID association from being completed) The user ID association should be deleted and then defined again.

DEFINE([*node*].*userid2*[/*password*]....)

specifies that a user ID association is to be formed between *userid1* at the node

where the command was issued, and *userid2* at *node*. If you specify more than one *node.userid2* operand, an association is established between *userid1* and each *node.userid2* specified. A user ID association enables RACF users to utilize command direction and password synchronization.

To issue the RACLINK DEFINE command, you need READ access to the following profile in the RRSFDATA class:

- RACLINK.DEFINE.node

The RRSFDATA class must be active.

When the DEFINE keyword is specified and the command issuer has sufficient authority to perform the RACLINK command for the user ID, the user ID association is implicitly approved if:

- A valid password is supplied for *node.userid2* on the DEFINE keyword.
- The command issuer has one of the following authorities over *userid2* on the DEFINE keyword:
 - SPECIAL
 - The profile is within the scope of a group in which the command issuer has the group-SPECIAL attribute
 - The command issuer is the owner of the profile.
- The command issuer has an association with a user ID on the node specified on the DEFINE keyword. That association must be either a PEER association or a MANAGED association with the command issuer as the manager. The user ID with which the command issuer has the association must have one of the following authorities over *userid2* on the DEFINE keyword:
 - SPECIAL
 - It is within the scope of a group that has the group-SPECIAL attribute
 - It is the owner of the profile

Otherwise, a user ID association requires explicit approval by *node.userid2* with the RACLINK APPROVE command.

Although it is possible for the command issuer to have more than 50 associated user IDs on the target node, only the first 50 are used for authority checking. RACLINK issues a message if more than 50 user ID associations exist for the command issuer.

An association is **pending** until *node.userid2* either approves the association with a RACLINK APPROVE command or rejects the association with a RACLINK UNDEFINE command.

Notes:

1. Under certain circumstances, RACLINK DEFINE(*node.userid*) requests can be issued by two users. If both requests are consistent, RACF treats this as an implicit approval. The entry is marked established in the target user IDs profile. An entry is considered consistent if the association type (PEER(PWSYNC) or PEER(NOPWSYNC)) is the same. If the request is not consistent (for example, differing PEER definitions or both users requesting a MANAGED association), RACF fails the request and the entries remain in a pending state. In order to correct this situation, the user(s) need to undefine and redefine the user ID associations.
2. When creating a user ID association with a revoked user ID:

RACLINK

- If a RACLINK DEFINE command is coded without the password operand and the target user ID is a revoked user, the results vary depending on the authority of the command issuer and the user ID associations of the command issuer. When:
The user ID association with the revoked user ID is created and the status displayed by a RACLINK LIST command is “ESTABLISHED” when one of the following is true:
 - The command issuer has sufficient authority (SPECIAL, group-SPECIAL, or owner) over the target user ID. **or**
 - The command issuer has a PEER association or is the manager of a MANAGED association with a user ID on the target node and the associated user ID has sufficient authority over the target user ID.
- 3. If a RACLINK DEFINE command is coded without the password operand, the target user ID is a revoked user ID, and the command issuer does not have sufficient authority (SPECIAL, group-SPECIAL, or owner) over the target user ID, the user ID association is created and the status displayed by a RACLINK LIST command is “PENDING APPROVAL BY *userid2*”.
- 4. If a RACLINK DEFINE command is coded with the password operand and the target user is a revoked user, the user ID association is not established and the status displayed by a RACLINK LIST is “SYSTEM ERROR”.

The type of association you want to establish is specified with one of the following:

MANAGED

specifies a managed association.

A managed association does not provide password synchronization. A managed association allows commands to be directed from the managing user ID to the managed user ID (that is, from *userid1* to *node.userid2*).

A managed association does not allow commands to be directed from the managed user ID to the managing user ID (that is, *node.userid2* cannot direct commands to *userid1*).

PEER(NOPWSYNC)

specifies a peer association without password synchronization.

Either user ID in a peer association can direct commands to the other user ID in the association.

If no association type is specified, PEER(NOPWSYNC) is the default.

PEER(PWSYNC)

specifies a peer association with password synchronization.

Either user ID in a peer association can direct commands to the other user ID in the association.

If either user in the association changes their password, the password is automatically changed for the other user in the association.

READ access to the the RACLINK.PWSYNC.node resource is required to use the RACLINK command to define a peer association with the PWSYNC attribute. READ access to the PWSYNC resource is required to synchronize the passwords when one of the associated users changes their password.

If the RRSFDATA class is not active, you cannot define an association with the PWSYNC attribute, or synchronize passwords.

UNDEFINE([node].userid2...)

specifies that a user ID association is ended between *userid2* on *node* and *userid1* on the node where the command is processed. Either member of an association can end an association.

If a user ID has attempted to establish an association with your user ID which requires approval, and you do not want to approve it, use the UNDEFINE keyword to reject the pending association.

APPROVE([node1].userid1...)

specifies that *userid2* on *node2* approves of a pending association between *userid2* at *node2* and *userid1* at *node1*. *node1* is the node where the RACLINK DEFINE was issued, and *node2* is the node where *userid2* issues the command.

Examples

Table 51. RACLINK Examples

Example 1

Operation The security administrator wants to know what, if any, associations user DENICE has with user BETH.

Known The security administrator wants to issue the command as a RACF TSO command.

Command RACLINK ID(DENICE) LIST(*.BETH)

Defaults None

Output See Figure 42.

Example 2

Operation User DENICE wants to define password synchronization between all of her MVS user IDs; DENICE at NODE1, DENICE at NODE2, and DENICE at NODE3.

Known DENICE wants to issue the command as a RACF TSO command. DENICE has the authority to issue the RACLINK command for her own user IDs and has the authority to establish password synchronization for her own user IDs. The command is to be issued from DENICE at NODE1.

Command RACLINK DEFINE(NODE2.DENICE/passw2 NODE3.DENICE/passw3) PEER(PWSYNC)

Defaults None

Results DENICE at NODE1 receives the following messages:

```

IRRT032I RACLINK command to associate user ID DENICE with
        NODE2.DENICE is pending approval.
IRRT032I RACLINK command to associate user ID DENICE with
        NODE3.DENICE is pending approval.
IRRP097I Peer association with DENICE at node NODE2 has been
        approved.
IRRP097I Peer association with DENICE at node NODE3 has been
        approved.
```

When user DENICE changes her password on one of her MVS user IDs, the new password propagates to take affect on her other user IDs. The password is checked for validity only on the node where user DENICE issues the command to change her password, not at any of the other nodes.

RACLINK

Table 51. RACLINK Examples (continued)

Example 3

Operation User BETH wants to define a MANAGED user ID association where BETH is the managing user ID and DENICE is the managed user ID.

Known User BETH:

- wants to issue the command as a RACF TSO command,
- does not know the password for user DENICE, and
- has the authority to issue the RACLINK command for her own user ID.

Command RACLINK DEFINE(NODE1.DENICE) MANAGED

Defaults None

Results User BETH receives the following message:

IRRT032I RACLINK command to associate user ID BETH with
NODE1.DENICE is pending approval.

User DENICE receives the following message:

IRRP094I Managed association with DENICE at node NODE1 issued
by BETH waiting for your approval.

The association remains pending until DENICE at NODE1 either approves the association with a RACLINK APPROVE command or rejects the association with a RACLINK UNDEFINE command.

ASSOCIATION information for user ID DENICE on node NODE1
at 1:12:31 on 04/01/95:

Association Type	Node.userid	Password Sync	Association Status
PEER OF	NODE1.BETH	YES	ESTABLISHED
MANAGED BY	NODE2.BETH	N/A	PENDING APPROVAL BY DENICE
PEER OF	NODE3.BETH	NO	PENDING APPROVAL BY BETH

Figure 42. Example 1: Output for the RACLINK LIST Command

RALTER (Alter General Resource Profile)

Purpose

Use the RALTER command to:

- Alter the profile for one or more resources belonging to classes defined in the class descriptor table. Using RALTER to modify an automatic TAPEVOL profile (a profile RACF creates automatically as part of protecting a tape data set) makes that TAPEVOL profile nonautomatic. For more information about TAPEVOL profiles, see *z/OS Security Server RACF Security Administrator's Guide*.
- Change the global access checking table
- Change the list of security categories
- Change the list of security levels

To have changes take effect after altering a generic profile if the class is RACLISTed using the RACROUTE REQUEST=LIST, GLOBAL=YES, or SETROPTS RACLIST, one of the following steps is required:

- The security administrator issues the SETROPTS command:

```
SETROPTS GENERIC(class-name) REFRESH
```

See the SETROPTS command for authorization requirements.

- The user of the resource logs off and logs on again.

To have changes take effect after altering a generic profile if the class has been RACLISTed, the security administrator issues the following command:

```
SETROPTS RACLIST(class-name) REFRESH
```

Attention:

- When the RALTER command is issued from ISPF, the TSO command buffer (including SESSKEY, SSIGNON and possible BINDPW password data) is written to the ISPLOG data set. As a result, you should not issue this command from ISPF or you must control the ISPLOG data set carefully.
- When the command is issued as a RACF operator command, the command (including SESSKEY, SSIGNON and possible BINDPW password data) is written to the system log. Therefore, if any of the sensitive operands are used the command should be issued via TSO, not as an operator command.

Issuing Options

The following table identifies the eligible options for issuing the RALTER command:

Table 52. How the RALTER Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	Yes	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

RALTER

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To define a general resource profile, see “RDEFINE (Define General Resource Profile)” on page 337.
- To list a general resource profile, see “RLIST (List General Resource Profile)” on page 382.
- To permit or deny access to a general resource profile, see “PERMIT (Maintain Resource Access Lists)” on page 247.
- To obtain a list of general resource profiles, see “SEARCH (Search RACF Database)” on page 408.

Authorization Required

When issuing the RALTER command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

To alter the profile for a resource belonging to a class defined in the class descriptor table, you must have sufficient authority over the resource. RACF makes the following checks until one of the conditions is met:

- You have the SPECIAL attribute.
- The resource profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You are the owner of the profile.
- If the profile is in the FILE or DIRECTRY class, the second qualifier of the profile name is your user ID.
- To assign a security label, you must have the SPECIAL attribute or have READ access to the security label profile. However, the security administrator can limit the ability to assign security labels to only users with the SPECIAL attribute.
- To assign a security category to a profile, you must have the SPECIAL attribute, or the access category must be in your user profile.
- To assign a security level to a profile, you must have the SPECIAL attribute, or, in your own profile, a security level that is equal to or greater than the security level you are assigning.
- To modify information in segments other than the base segment such as DLFDATA, SPECIAL or field-level access checking is required.

For discrete profiles only:

- You are on the access list for the resource and you have ALTER authority. If you have any other level of authority, you cannot use the command for this resource.
- Your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is in the access list and has ALTER authority.
- The universal access authority for the resource is ALTER.

For both discrete and generic profiles, when you specify the GLOBALAUDIT operand:

- You have the AUDITOR attribute or the profile is within the scope of a group in which you have group-AUDITOR attribute.

The following operands have restrictions noted with the description of each operand:

- ADDMEM
- DELMEM
- ADDVOL
- GLOBALAUDIT

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the RALTER command is:

```
[subsystem-prefix]{RALTER | RALT}

    class-name
    (profile-name ...)
    [ ADDCATEGORY(category-name ...)
    | DELCATEGORY [ ( {category-name...
    | * } ) ] ]
    [ {ADDMEM | DELMEM} (member ...) ]
    [ {ADDVOL | DELVOL} (volume-serial ...) ]
    [ APPLDATA('application-data') | NOAPPLDATA ]
    [ AT( [ node ] .userid ...) | ONLYAT( [ node ] .userid ...) ]
    [ AUDIT( access-attempt [ (audit-access-level) ] ...) ]
    [ DATA('installation-defined-data') | NODATA ]
    [ DLFDATA(
        [ RETAIN(YES|NO) | NORETAIN ]
        [NOJOBNAMES
            | JOBNAMES(jobname1 ...)
            | ADDJOBNAMES(jobname1 ...)
            | DELJOBNAMES(jobname1 ...) ] )
    | NODLFDATA ]
    [ EIM(
        [DOMAINDN(eim_domain_dn) | NODOMAINDN]
        [OPTIONS (ENABLE | DISABLE) | NOOPTIONS]
        [LOCALREGISTRY(registry_name) | NOLOCALREGISTRY])
    | NOEIM ]
    [ GLOBALAUDIT(access-attempt [ (audit-access-level) ] ...) ]
```

RALTER

```
[ KERB (
  [ DEFTKTLFE(def-ticket-life) | NODEFTKTLFE ]
  [ ENCRYPT (
    [ DES | NODES ]
    [ DES3 | NODES3 ]
    [ DESD | NODESD ] )
  | NOENCRYPT ]
  [ KERBNAME(kerberos-realm-name)
  | NOKERBNAME ]
  [ MAXTKTLFE(max-ticket-life) | NOMAXTKTLFE ]
  [ MINTKTLFE(min-ticket-life) | NOMINTKTLFE ]
  [ PASSWORD(kerberos-password) | NOPASSWORD ] )
| NOKERB ]
[ LEVEL(nn) ]
[ NOTIFY [(userid)] | NONOTIFY ]
[ OWNER(userid or group-name) ]
[ PROXY(
  [ LDAPHOST(ldap_url) | NOLDAPHOST ]
  [ BINDDN(bind_distinguished_name) | NOBINDDN ]
  [ BINDPW(bind_password) | NOBINDPW ] )
| NOPROXY ]
[ SECLABEL(seclabel-name ...) | NOSECLABEL ]
[ SECLEVEL(seclabel-name ...) | NOSECLEVEL ]
[ SESSION(
  [ CONVSEC( NONE | CONV | ALREADYV
              | PERSISTV | AVPV )
  | NOCONVSEC ]
  [ INTERVAL(n) | NOINTERVAL ]
  [ LOCK | NOLOCK ]
  [ SESSKEY(session-key) | NOSESSKEY ] )
| NOSESSION ]
[ SINGLEDSN | NOSINGLEDSN ]
[ SSIGNON(
  [ KEYMASKED(key-value) |
  [ KEYENCRYPTED(key-value)] ] )
| NOSSIGNON ]
[ STDATA(
  [ USER(userid | =MEMBER) | NOUSER ]
  [ GROUP(group-name | =MEMBER) | NOGROUP ]
  [ PRIVILEGED( NO | YES) | NOPRIVILEGED ]
  [ TRACE( NO | YES) | NOTRACE ]
  [ TRUSTED( NO | YES) | NOTRUSTED ] )
| NOSTDATA ]
[ SVFMR(
  [ SCRIPTNAME(script-name) | NOSCRIPTNAME ]
  [ PARMNAME(parm-name) | NOPARMNAME ] )
| NOSVFMR ]
[ TIMEZONE( {E | W} hh [ .mm ] ) | NOTIMEZONE ]
```

```
[ TME(
  [ CHILDREN(profile-name ...)
  | ADDCHILDREN(profile-name ...)
  | DELCHILDREN(profile-name ...)
  | NOCHILDREN ]
  [ GROUPS(group-name ...)
  | ADDGROUPS(group-name ...)
  | DELGROUPS(group-name ...)
  | NOGROUPS ]
  [ PARENT(profile-name)
  | NOPARENT ]
  [ RESOURCE(resource-access-specification ...)
  | ADDRESOURCE(resource-access-specification ...)
  | DELRESOURCE(resource-access-specification ...)
  | NORESOURCE ]
  [ ROLES(role-access-specification ...)
  | ADDROLES(role-access-specification ...)
  | DELROLES(role-access-specification ...)
  | NOROLES ] )
| NOTME ]
[ TVTOC | NOTVTOC ]
[ UACC(access authority) ]
[ WARNING | NOWARNING ]
[ WHEN( [ DAYS(day-info) ] [ TIME(time-info) ] ) ] ]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

class-name

specifies the name of the class to which the resource belongs. Valid class names are those defined in the class descriptor table supplied by IBM or in the installation-defined class descriptor table. For a list of general resource classes supplied by IBM, see Appendix B, “Description of RACF Classes” on page 507.

This operand is required and must be the first operand following RALTER.

This command is not intended to be used for profiles in the following classes:

- DCEUIDS
- DIGTCERT
- DIGTNMAP
- DIGTRING

RALTER

- NDSLINK
- NOTELINK
- ROLE
- UNIXMAP

(*profile-name* ...)

specifies the name of the profile you want to change. The name you specify must be the name of an existing discrete or generic profile in the specified class. RACF uses the class descriptor table to determine the syntax of resource names within the class and whether the resource is a group.

Mixed-case profile names are accepted and preserved when *class-name* refers to a class defined in the class descriptor table with CASE=ASIS.

This operand is required and must be the second operand following RALTER.

Notes:

1. For class TAPEVOL, if the volume serial specified for *profile-name* is a member of a tape volume set, then the profile definition for all tapes in the set is changed, since there is only one profile for the tape volume set.
A tape volume set is used to refer to a set of two or more tapes created by the overflow of one tape to the next. RACF protects these tapes with one profile. Hence, if the value specified for *profile-name* on this command is a member of a tape volume set, the changes in its resource profile affect the other members of the set.
2. You can specify only a single volume serial number if you also specify the ADDVOL or DELVOL operand.
3. To define a controlled program, you must specify *class-name* as PROGRAM and also specify ADDMEM or DELMEM. Also, you can specify only one *profile-name*.
4. If you specify *class-name* as PROGRAM, *profile-name* must identify one or more load modules or program objects. If you specify the full name of the program, the profile applies only to load modules or program objects with that specific name. If you specify the last character of the name as an *, the profile applies to all load modules or program objects that match the preceding part of the name, but only if they reside in one of the libraries listed in the profile's member list. For example, IKF* identifies all load module names that begin with IKF. If you specify *profile-name* as * or **, then the profile applies to all load modules and program objects that reside in one of the libraries you identify in the profile's member list, unless a profile with a more specific name and matching library applies.
5. For Security Server Network Authentication Service, the profile name for the definition of the local realm must be KERBDFLT.
6. RACF processes each profile name you specify independently, and all operands you specify apply to each named profile name. If an error occurs while processing a profile name, RACF issues a message and continues processing with the next profile name.

ADDCATEGORY | DELCATEGORY

ADDCATEGORY(*category-name* ...)

specifies one or more names of installation-defined security categories. The *category-name* you specify must be defined as members of the CATEGORY profile in the SECDATA class. (For information on defining security categories, see *z/OS Security Server RACF Security Administrator's Guide*.)

Specifying ADDCATEGORY causes RACF to add any *category-names* you specify to any list of required categories that already exists in the resource profile. All users previously allowed to access the resource can continue to do so only if their profiles also include the additional values for *category-names*.

When the SECDATA class is active and you specify ADDCATEGORY, RACF performs security category checking in addition to its other authorization checking. If a user requests access to a resource, RACF compares the list of security categories in the user profile with the list of security categories in the resource profile. If RACF finds any security category in the resource profile that is not in the user's profile, RACF denies access to the resource. If the user's profile contains all the required security categories, RACF continues with other authorization checking.

Note: RACF does not perform security category checking for a started task with the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF started procedures table or STARTED class. Also, RACF does not enforce security category information specified on profiles in the PROGRAM class.

DELCATEGORY[(*category-name ...!)]**

specifies one or more names of installation-defined security categories you want to delete from the resource profile. Specifying an asterisk (*) deletes all categories; RACF no longer performs security category checking for the resource.

Specifying DELCATEGORY by itself causes RACF to delete from the profile only undefined category names (those category names that were once known to RACF but that the installation has since deleted from the CATEGORY profile).

ADDMEM | DELMEM

Specifies the resource names that RACF is to add to, or delete from, the member list of the resource group profile indicated by *profile-name*.

ADDMEM(*member....*)

You can use the ADDMEM operand to perform tasks such as altering security categories and security levels, entries in the global access checking table, and entries for program control as described in the following sections.

Mixed-case member names are accepted and preserved when *class-name* refers to a class defined in the class descriptor table with CASE=ASIS. When *class-name* is GLOBAL and *profile-name* is the name of a class defined in the class descriptor table with CASE=ASIS, the name part of a member entry in the GLOBAL access table is preserved as entered.

When specifying the &RACUID keyword with ADDMEM, generic characters such as the asterisk (*) and the percent sign (%) cannot follow the keyword.

To add members using the RALTER command, you need one of the following authorities, in addition to the authority needed to issue the RALTER command:

1. For classes other than PROGRAM, SECADATA, GLOBAL, RACFVARS, and NODES, if the member resources are already RACF-protected by a member class profile or as a member of a profile in the same grouping class, one of the following must be true:

RALTER

- You have ALTER access authority to the member.
 - You are the owner of the member resource.
 - The member resource is within the scope of a group in which you have the group-SPECIAL attribute.
 - You have the SPECIAL attribute.
2. For classes other than PROGRAM, SECDATA, GLOBAL, RACFVARS, and NODES, if the member resources are not RACF-protected (that is, there is no profile defined for that member), one of the following must be true:
 - You have CLAUTH authority to define resources in the member resource class.
 - You have the SPECIAL attribute.
 3. To add a member to a profile in the RACFVARS or NODES class, one of the following must be true:
 - You have CLAUTH authority to define resources in the specified class (for example, RACFVARS or NODES).
 - You have the SPECIAL attribute.
 - You are the owner of the profile indicated by *profile-name*.
 - You have ALTER access authority to the profile indicated by *profile-name*.
 4. To add a member to a profile in the PROGRAM or SECDATA class, one of the following must be true:
 - You have CLAUTH authority to define resources in the specified class (for example, PROGRAM or SECDATA).
 - You have the SPECIAL attribute.
 5. To add a member to a profile in the GLOBAL class (other than the GLOBAL DATASET, GLOBAL DIRECTRY, or GLOBAL FILE profile) where the syntax is:

```
RALT GLOBAL class-name ADDMEM(resource-name/access-level)
```

one of the following must be true:
 - If the profile *resource-name* is already RACF-protected by a profile in class *class-name*:
 - You have ALTER access authority to the profile *resource-name* in class *class-name*.
 - You are the OWNER of the profile *resource-name*.
 - The profile *resource-name* in class *class-name* is within the scope of a group in which you have the group-special attribute.
 - You have the SPECIAL attribute.
 6. If the profile *resource-name* is not already RACF-protected (that is, there is no profile defined for that member in class *class-name*):
 - You have CLAUTH authority to define resources in the class *class-name*.
 - You have the SPECIAL attribute.
 7. To add a member to the GLOBAL DATASET profile, one of the following must be true:
 - You are the owner of the DATASET profile in the GLOBAL class.

- The member is within the scope of a group in which you have the group-SPECIAL attribute, or the high-level qualifier of the member name is your userid.
 - You have the SPECIAL attribute.
8. To add a member to the GLOBAL DIRECTRY or GLOBAL FILE profile, you must have the SPECIAL attribute.

For more information on the format of member names in general, and for specific classes (GLOBAL, NODES, PROGRAM, SECDATA) see 346.

DELMEM(*member....*)

specifies the resource names that are to be deleted from the resource group indicated by *profile-name*. This operand is ignored if the class name specified is not a resource group class.

If *class-name* is specified as GLOBAL or PROGRAM, the rules for “member” are the same as given for ADDMEM. If *class-name* is specified as SECDATA, “member” should be a valid SECLEVEL name or category name.

Mixed-case member names are accepted and preserved when *class-name* refers to a class defined in the class descriptor table with CASE=ASIS. When *class-name* is GLOBAL and *profile-name* is the name of a class defined in the class descriptor table with CASE=ASIS, the name part of a member entry in the GLOBAL access table is preserved as entered.

ADDVOL | DELVOL

ADDVOL(*volume-serial ...*)

specifies the tape volume serial numbers to be added to the tape volume set represented by *profile-name*. When you specify ADDVOL, *profile-name* must be a single volume serial number, which can identify any of the volumes currently defined in the volume set.

To use the ADDVOL operand, you must have the SPECIAL attribute, or you must have the CLAUTH attribute for the TAPEVOL resource class in addition to the other RACF requirements for using the RALTER command.

If you specify a generic profile, RACF ignores this operand.

Notes:

1. The ADDVOL operand is only valid for the TAPEVOL resource class.
2. If you specify both ADDVOL and DELVOL, RACF uses the last operand that you specify.

DELVOL(*volume-serial ...*)

specifies the tape volume serial numbers to be deleted from the tape volume set represented by *profile-name*. When you specify DELVOL, *profile-name* must be a single volume serial number, which can identify any of the volumes currently defined in the volume set except one of the volumes to be deleted. If you specify the same volume serial number for both *profile-name* and DELVOL, RACF ignores it.

If you try to delete a tape volume when the TAPEVOL profile contains one or more TVTOC entries, RACF does not complete the command if a TVTOC entry indicates that there is a protected data set on the volume. To delete this volume, you must first use the DELDSD command to delete any protected data sets on the volume.

If you specify a generic profile, RACF ignores this operand.

RALTER

Notes:

1. The DELVOL operand is only valid for the TAPEVOL resource class.
2. If you specify both ADDVOL and DELVOL, RACF uses the last operand that you specify.

APPLDATA | NOAPPLDATA

APPLDATA('application-data')

Specifies a text string that is associated with each of the named resources. The text string can contain a maximum of 255 characters and must be enclosed in single quotes. It can also contain double-byte character set (DBCS) data.

Rules:

- For profiles in the PROGRAM class, RACF will examine the APPLDATA (if any) and perform special processing if you have specified 'MAIN' or 'BASIC' (optionally followed by blanks). This processing will occur only for profiles whose names do not end in *, and only when you have enabled enhanced PGMSECURITY mode. For details of this processing, see *z/OS Security Server RACF Security Administrator's Guide*.
- For the FACILITY class, RACF examines the APPLDATA value for these profiles:
 - BPX.DEFAULT.USER, the APPLDATA specifies a user ID and group name from which RACF can retrieve default OMVS segment information.
 - BPX.NEXT.USER, the APPLDATA specifies information that RACF will use for the automatic assignment of OMVS UIDs and GIDs.
 - IRR.PGMSECURITY, the APPLDATA specifies whether RACF will operate in basic, enhanced, or enhanced-warning PGMSECURITY mode.
 - If the APPLDATA is exactly 'ENHANCED' then RACF will run in enhanced PGMSECURITY mode.
 - If the APPLDATA is exactly 'BASIC' then RACF will run in basic PGMSECURITY mode
 - If the APPLDATA is empty or contains any other value, RACF will run in enhanced PGMSECURITY mode but in warning mode rather than failure mode.
- For the TIMS and GIMS class, specify *application-data* as REVERIFY to force the user to reenter his password whenever the transaction or transactions listed in the *profile-name* or ADDMEM operands are used.
- For the PTKTDATA class, the *application-data* field can be used to control the replay protection function of PassTicket support.
 - PassTicket replay protection prevents the use of user IDs to be shared among multiple users. However, in some events it is desirable to bypass this replay protection function.
 - Specifying "no replay protection" in the *application-data* field indicates that replay protection is to be bypassed. For example:

```
RALTER PTKTDATA profile-name APPLDATA('NO REPLAY PROTECTION')
```


would result in replay protection being bypassed.

Note the following:

- There **must** be a single space between the words "no" and "replay", and "replay" and "protection". Lack of spaces **or** additional spaces or characters will make the command ineffective. For example, entering

```
RALTER PTKTDATA profile-name APPLDATA('NOREPLAY PROTECTION')
```

would result in replay protection not being bypassed.

- The text string 'no replay protection' will always be rolled to upper case.
- The text string 'no replay protection' can appear anywhere in the APPLDATA field.
- See *z/OS Security Server RACF Security Administrator's Guide* for more information on the PassTicket function.
- RACF will not validate the APPLDATA during RALTER and will not issue any messages during the subsequent processing if it finds an unexpected value.
- This information, if present, can be displayed with the RLIST command. See *z/OS Security Server RACF Security Administrator's Guide* for more information on these APPLDATA values.

NOAPPLDATA

specifies that the RALTER command is to delete the text string that was present in the profile associated with the resource.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([*node*].*userid* ...)

specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT([*node*].*userid* ...)

specifies that the command is to be directed only to the node specified by *node* where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed only to the local node.

AUDIT(*access-attempt*[(*audit-access-level*)])

access-attempt

specifies which access attempts you want to log on the SMF data set. The following options are available:

ALL

specifies that you want to log both authorized accesses and detected unauthorized attempts to access the resource.

FAILURES

specifies that you want to log detected unauthorized attempts to access the resource.

NONE

specifies that you do not want any logging to be done for accesses to the resource.

RALTER

SUCCESS

specifies that you want to log authorized accesses to the resource.

audit-access-level

specifies which access levels you want to log on the SMF data set. The levels you can specify are:

ALTER

logs ALTER access-level attempts only.

CONTROL

logs access attempts at the CONTROL and ALTER levels.

READ

logs access attempts at any level. This is the default value if no access level is specified.

UPDATE

logs access attempts at the UPDATE, CONTROL, and ALTER levels.

You cannot audit access attempts at the EXECUTE level.

DATA | NODATA

DATA(*'installation-defined-data'*)

specifies up to 255 characters of installation-defined data to be stored in the profile for the resource. The data must be enclosed in single quotes.

This information is listed by the RLIST command.

NODATA

specifies that the RALTER command is to delete the installation-defined data in the resource profile.

DLFDATA | NODLFDATA

DLFDATA

for profiles in the DLFCLASS, specifies information used in the control of DLF objects.

RETAIN(YES|NO) | NORETAIN

specifies whether the DLF object can be retained after use.

NOJOBNAMES | JOBNAMES | ADDJOBNAMES | DELJOBNAMES

You can specify any job name valid on your system. You can also specify generic job names with an asterisk (*) as the last character of a job name, to indicate generic job names. For example, JOBNAMES(ABC) allows only job ABC to access the DLF objects protected by the profile. JOBNAMES(ABC*) allows any job whose name begins with ABC (such as ABC, ABC1, or ABCDEF and so forth) to access to the DLF objects.

NOJOBNAMES

specifies that no job names can access the DLF objects protected by this profile.

JOBNAMES(*jobname1.....*)

specifies the list of job names that can access the DLF objects protected by this profile.

ADDJOBNAMES(*jobname1.....*)

adds to the list of job names, the job names that can access the DLF objects protected by this profile.

DELJOBNAMES(*jobname1.....*)

deletes the names from the job names list.

NODLFDATA

deletes the DLFDATA in the specified segment

EIM | NOEIM

The EIM keyword with the DOMAINDN, OPTIONS, and LOCALREGISTRY subkeywords combined with the PROXY segment define the EIM domain, LDAP host it resides on, and bind information required by the EIM services to establish a connection with an EIM domain. The EIM services will attempt to retrieve this information when it is not explicitly supplied via invocation parameters.

DOMAINDN(*eim_domain_dn*)|NODOMAINDN

DOMAINDN(*eim_domain_dn*)

Specifies the distinguished name of the EIM domain. A valid EIM domain distinguished name begins with `ibm-eimDomainName=`. Uppercase and lowercase are accepted and maintained in the case in which they are entered. The EIM domain distinguished name is one component of an EIM domain name. An EIM domain name identifies the LDAP server that stores the EIM domain information. The EIM domain name begins with the *ldap_url* from the LDAPHOST suboperand of the PROXY keyword, followed by "/" and ends with the *eim_domain_dn* from the DOMAINDN suboperand. The length of a valid EIM domain name is determined by the combination of those factors. RACF allows the input of 1023 characters for the domain distinguished name. RACF does not ensure that an EIM domain name created from the LDAP URL and EIM domain distinguished name forms a valid EIM domain name. See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP distinguished names.

NODOMAINDN

Deletes the *eim_domain_dn*.

OPTIONS

specifies options that control the EIM configuration.

ENABLE|DISABLE

ENABLE

Specifies that new connections may be established with the specified EIM domain. This is the default.

DISABLE

Specifies that new connections may not be established with the specified EIM domain.

NOOPTIONS

Resets OPTIONS to the default value of ENABLE.

LOCALREGISTRY(*registry_name*)

Specifies the name of the local RACF registry in EIM domains. This operand is only valid for the IRR.PROXY.DEFAULTS FACILITY class profile. The value is ignored when used on other profiles. Changing this value does not affect the in-storage copy of the registry name until SETROPTS EIMREGISTRY is issued or the system is IPLed. The local *registry_name* may be 1 to 255 characters long and is not case sensitive. The local registry name is stored in the RACF database in uppercase.

RALTER

NOLOCALREGISTRY

Deletes the local registry name from the profile. It does not affect the in-storage copy of the registry name. Issue SETROPTS NOEIMREGISTRY or IPL the system to remove the in-storage copy.

NOEIM

Deletes the EIM segment.

GLOBALAUDIT(*access-attempt*[(*audit-access-level*)])

access-attempt

specifies which access attempts the user who has the AUDITOR attribute wants to log on the SMF data set.

ALL

specifies that you want to log both authorized accesses and detected unauthorized attempts to access the resource.

FAILURES

specifies that you want to log detected unauthorized attempts to access the resource.

NONE

specifies that you do not want any logging to be done for accesses to the resource.

SUCCESS

specifies that you want to log authorized accesses to the resource.

audit-access-level

specifies which access levels the user who has the AUDITOR attribute wants to log on the SMF data set.

ALTER

logs ALTER access-level attempts only.

CONTROL

logs access attempts at the CONTROL and ALTER levels.

READ

logs access attempts at any level. This is the default value if no access level is specified.

UPDATE

logs access attempts at the UPDATE, CONTROL, and ALTER levels.

You cannot audit access attempts at the EXECUTE level.

To use GLOBALAUDIT, you must have the AUDITOR attribute, or the resource profile must be within the scope of a group in which you have the group-AUDITOR attribute.

Regardless of the value you specify for GLOBALAUDIT, RACF always logs all access attempts specified on AUDIT.

KERB | NOKERB

KERB

specifies Security Server Network Authentication Service information for a REALM class profile.

DEFTKTLFE(*def-ticket-life*) | NODEFTKTLFE

DEFTKTLFE(*def-ticket-life*)

specifies the default ticket lifetime for the local Network Authentication Service in seconds. DEFTKTLFE is a numeric value between 1 and 2 147 483 647. Note that 0 is not a valid value.

This keyword is only applicable when defining the KERBDFLT REALM profile for the local realm.

The RALTER command only requires specification of all of the ticket lifetime keywords on the same command invocation if RALTER is being used to initially define these values. If values have been previously defined, RACF uses both the previous values and new values specified to verify the specified *def-ticket-life* value.

NODEFTKTLFE

deletes the *def-ticket-lifetime* value for the local Network Authentication Service.

ENCRYPT [[(DES | NODES) [DES3 | NODES3] [DESD | NODESD)]]
NOENCRYPT

The ENCRYPT values are used to specify which keys are allowed for use based on the encryption algorithm used to generate them. The default values for ENCRYPT are DES, DES3, and DESD. You can use the following values to specify which keys are allowed for use by a principal.

DES	DES encrypted keys are allowed for use.
NODES	No DES encrypted keys are allowed for use.
DES3	DES3 encrypted keys are allowed for use.
NODES3	No DES3 encrypted keys are allowed for use.
DESD	DESD encrypted keys are allowed for use.
NODESD	No DESD encrypted keys are allowed for use.

The values in effect are dependent on the current SETROPTS KERBLVL setting.

When SETROPTS KERBLVL(0) is in effect the ENCRYPT settings will be ignored. Regardless of the settings DES keys will be generated and processed.

When SETROPTS KERBLVL(1) is in effect, or when SETROPTS KERBLVL gets changed from 0 to 1, the ENCRYPT settings will go into effect. Therefore, on password change, all three keys are generated and stored in the user's profile. The ENCRYPT setting will be used to determine which keys can be processed.

If you do not want to accept the defaults, you must specify the values you desire. For example, if you want to use only DES3 encryption, you must specify ENCRYPT(NODES DES3 NODESD).

If you specify ENCRYPT(NODES, NODES3, NODESD) at KERBLVL(1), no keys can be used, but all three will be generated and stored. At KERBLVL(0), the DES key will still be generated and it cannot be disallowed.

NOENCRYPT

Specifies that there is no restriction on which generated keys are to be allowed, and resets KERB encryption to the default settings. The NOENCRYPT operand has no effect at KERBLVL(0).

KERBNAME(*kerberos-realm-name*) | NOKERBNAME

KERBNAME(*kerberos-realm-name*)

specifies the unqualified name of the local realm for Network Authentication Service. The maximum length of this field is 117 characters. The fully qualified form of the local realm name

/.../kerberos_realm_name/krbtgt/kerberos_realm_name

must not be specified.

The name assigned to the local realm limits the length of local principal names, since fully qualified local principal names

/.../kerberos_realm_name/principal_name

cannot exceed 240 characters.

The local realm name that you define to RACF can consist of any character, except the / (X'61') character. It is highly recommended that you avoid using **any** of the EBCDIC variant characters to prevent problems with different code pages. You can enter the name with or without single quotes, depending on the following:

- If parentheses, commas, blanks, or semicolons are entered as part of the name, the character string must be enclosed in single quotes.
- If a single quote is intended to be part of the name and the entire character string is enclosed in single quotes, you must use two single quotes together to represent each single quote within the string.
- If the first character of the name is a single quote, you must enter the string within single quotes, with two single quotes entered for the single quote.

Regardless of the case in which it is entered, RACF rolls the name of the local Network Authentication Service realm to upper case. However, RACF does not ensure that a valid *kerberos-realm-name* has been specified.

Note: Because of the relationship between the realm name and the local principal name where the length of a fully qualified name cannot exceed 240 characters, caution and planning must go into renaming the local realm since the combined length is only checked by RACF when a local *kerberos-principal-name* is added or altered. Renaming the realm should be avoided as a result. In addition, if the name of the realm does change, the user's keys will become unusable.

This keyword is only applicable when defining the KERBDFLT REALM profile for the local Network Authentication Service realm.

NOKERBNAME

deletes the *kerberos-realm-name*.

MAXTKTLFE(*max-ticket-life*) | NOMAXTKTLFE

MAXTKTLFE(*max-ticket-life*)

specifies the *max-ticket-life* for the local Security Server Network

RALTER

Authentication Service in seconds. MAXTKTLFE is a numeric value between 1 and 2 147 483 647. Note that 0 is not a valid value.

This keyword is only applicable when defining the KERBDFTL REALM profile for the local Network Authentication Service realm.

The RALTER command only requires specification of all of the ticket lifetime keywords on the same command invocation if RALTER is being used to initially define these values. If values have been previously defined, RACF uses both these previous values and new values specified on the command, to verify the specified *max-ticket-life* value.

NOMAXTKTLFE

deletes the *max-ticket-lifetime* value for the local Network Authentication Service.

MINTKTLFE(*max-ticket-life*) | NOMINTKTLFE

MINTKTLFE(*min-ticket-life*)

specifies the *min-ticket-life* for the Network Authentication Service in seconds. MINTKTLFE is a numeric value between 1 and 2 147 483 647. Note that 0 is not a valid value.

This keyword is only applicable when defining the KERBDFTL REALM profile for the local realm.

The RALTER command only requires specification of all of the ticket lifetime keywords on the same command invocation if RALTER is being used to initially define these values. If values have been previously defined, RACF uses both the previous values and new values specified on the command to verify the specified *min-ticket-life* value.

NOMINTKTLFE

deletes the *min-ticket-lifetime* value for the local Network Authentication Service principal.

PASSWORD(*kerberos-password*) | NOPASSWORD

PASSWORD(*kerberos-password*)

Specifies the value of the *kerberos-password*. The maximum length of this value is 8 characters. The PASSWORD keyword is applicable to all REALM class profile definitions. A password is required in order for the local realm to grant ticket-granting tickets and a password must be associated with the definition of an inter-realm trust relationship, or the definition is incomplete.

The *kerberos-password* that you define to RACF might consist of any character. It is highly recommended that use of **any** of the EBCDIC variant characters be avoided to prevent problems with different code pages. You can enter a password with or without single quotes, depending on the following:

- If parentheses, commas, blanks, or semicolons are entered as part of the password, the character string must be enclosed in single quotes.
- If a single quote is intended to be part of the password and the entire character string is enclosed in single quotes, you must use two single quotes together for each single quote within the string.

RALTER

- If the first character of the password is a single quote, you must enter the string within single quotes, with two single quotes entered for the character.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered.

Note: This keyword is intended for administrators to be able to associate a password with the definition of a realm. It is not the same as a RACF user password and is not constrained by the SETROPTS password rules and password interval values that might be established for RACF user passwords.

NOPASSWORD

deletes the Network Authentication Service password. If this is the local Network Authentication Service realm (KERBDFLT), it will no longer be able to grant ticket-granting tickets. Removal of the password from a foreign realm definition will invalidate the inter-realm trust relationship.

NOKERB

deletes the KERB segment.

LEVEL(*nn*)

specifies a level indicator, where *nn* is an integer between 00 and 99. Your installation assigns the meaning of the value. It is included on all records that log resource accesses and is listed by the RLIST command.

NOTIFY | NONOTIFY

NOTIFY[(*userid*)]

specifies the user ID of a RACF-defined user to be notified whenever RACF uses this profile to deny access to a resource. If you specify NOTIFY without specifying a user ID, RACF takes your user ID as the default; you are notified whenever the profile denies access to a resource.

If you receive NOTIFY messages, you should log on frequently to take action in response to the unauthorized access attempt described in each message. RACF sends NOTIFY messages to the SYS1.BROADCAST data set. When the resource profile also includes WARNING, RACF might have granted access to the resource to the user identified in the message.

When RACF denies access to a resource, it does **not** notify a user:

- When the resource is in the PROGRAM class
- When the resource is in a class for which an application has built in-storage profiles using RACROUTE REQUEST=LIST

Some applications, such as IMS™ and CICS, load all the profiles for a given class into storage. After these profiles are in storage, the applications can do a “fast” authorization check using RACROUTE REQUEST=FASTAUTH. Fast authorization checking is different from normal authorization checking in several ways. One difference is that, in some cases, fast authorization checking does not issue warning messages, notification messages or support auditing. In cases where it does not, return and reason codes are returned to the application to allow support of these functions. The application can examine the return and reason codes and use RACROUTE REQUEST=AUTH to create the messages and audit records. If the application uses RACROUTE

REQUEST=AUTH to support auditing, the specified user is notified. Otherwise, notification, warning, and such do not occur.

For details on using RACF with IMS, see *z/OS Security Server RACF Security Administrator's Guide*. For details on using RACF with CICS, refer to *CICS RACF Security Guide*.

- When the profile is used to disallow the creation or deletion of a data set NOTIFY is used only for resource access checking, not for resource creation or deletion.

NONOTIFY

specifies that no user is to be notified when RACF uses this profile to deny access to a resource.

OWNER(userid or group-name)

specifies a RACF-defined user or group to be assigned as the new owner of the resource you are changing.

To change the owner of a resource, you must be the current owner of the resource or have the SPECIAL attribute, or the profile must be within the scope of a group in which you have the group-SPECIAL attribute. The user specified as the owner does not automatically have access to the resource. Use the PERMIT command to add the owner to the access list as desired.

PROXY | NOPROXY

PROXY

specifies information which the z/OS LDAP Server will use when acting as a proxy on behalf of a requester. The R_proxyserv (IRRSPY00) SAF callable service will attempt to retrieve this information when it is not explicitly supplied via invocation parameters. Applications or other services which use the R_proxyserv callable service, such as IBM Policy Director Authorization Services for z/OS and OS/390, may instruct their invokers to define PROXY segment information.

LDAPHOST(ldap_url)

specifies the URL of the LDAP server which the z/OS LDAP Server will contact when acting as a proxy on behalf of a requester. An LDAP URL has a format such as ldap://12.34.56.78:389 or ldaps://12.34.56.78:636, where ldaps indicates that an SSL connection is desired for a higher level of security. LDAP will also allow you to specify the host name portion of the URL using either the text form (BIGHOST.POK.IBM.COM) or the dotted decimal address (12.34.56.78). The port number is appended to the host name, separated by a colon ':' (X'7A'). See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP URLs and how to enable LDAP servers for SSL connections.

The LDAP URL that you define to RACF can consist of 10—1023 characters. A valid URL must start with either ldap:// or ldaps://. RACF will allow any characters to be entered for the remaining portion of the URL, but you should ensure that the URL conforms to TCP/IP conventions. For example, parentheses, commas, blanks, semicolons, and single quotes are not typically allowed in a host name. The LDAP URL can be entered with or without single quotes, however, in both cases, it will be folded to upper case.

RACF does not ensure that a valid LDAP URL has been specified.

RALTER

NOLDAPHOST

deletes the URL of the LDAP server which the z/OS LDAP Server will contact when acting as a proxy on behalf of a requester.

BINDDN(*bind_distinguished_name*)

specifies the distinguished name (DN) which the z/OS LDAP Server will use when acting as a proxy on behalf of a requester. This DN will be used in conjunction with the BIND password, if the z/OS LDAP Server needs to supply an administrator or user identity to BIND with another LDAP Server. A DN is made up of attribute:value pairs, separated by commas. For example:

```
cn=Ben Gray,ou=editing,o=New York Times,c=US
cn=Lucille White,ou=editing,o=New York Times,c=US
cn=Tom Brown,ou=reporting,o=New York Times,c=US
```

See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP DNs.

When you define a BIND DN to RACF, it can contain 1-1023 characters. The BIND DN can consist of any characters and can be entered with or without single quotes. The following rules apply:

- If parentheses, commas, blanks, or semicolons are to be entered as part of the BIND DN, the character string must be enclosed in single quotes.
- If a single quote is intended to be part of the BIND DN, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered. See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP distinguished names.

If you issue the RALTER command as a RACF operator command and you specify the BIND DN in lowercase, you must include the BIND DN within single quotations.

RACF does not ensure that a valid BIND DN has been specified.

NOBINDDN

deletes the distinguished name (DN) used by the z/OS LDAP server when acting as a proxy on behalf of a requester.

BINDPW

specifies the password which the z/OS LDAP Server will use when acting as a proxy on behalf of a requester.

When you define a BIND password to RACF, it can contain 1-128 characters. The BIND password can consist of any characters (see exception below) and can be entered with or without single quotes. The following rules apply:

- The BIND password can not start with a left curly brace '{' (X'8B').
- If parentheses, commas, blanks, or semicolons are to be entered as part of the BIND password, the character string must be enclosed in single quotes.

RALTER

- If a single quote is intended to be part of the BIND password, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered. See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP passwords.

If you issue the RALTER command as a RACF operator command and you specify the BIND password in lowercase, you must include the BIND password within single quotations.

RACF does not ensure that a valid BIND password has been specified.

NOBINDPW

deletes the password used by the z/OS LDAP server when acting as a proxy on behalf of a requester.

NOPROXY

deletes LDAP proxy information.

SECLABEL | NOSECLABEL

SECLABEL(*seclabel-name*)

specifies an installation-defined security label for this profile. A security label corresponds to a particular security level (such as CONFIDENTIAL) with a set of zero or more security categories (such as PAYROLL or PERSONNEL).

If you are authorized to use the SECLABEL, RACF stores the name of the security label you specify in the resource profile.

If you are not authorized to the SECLABEL or if the name you had specified is not defined as a SECLABEL profile in the SECLABEL class, the resource profile is not updated. If the SECLABEL class is active and the security level is specified in this profile, any security levels and categories in the profile are ignored.

NOSECLABEL

removes the security label, if one had been specified, from the profile.

SECLEVEL | NOSECLEVEL

SECLEVEL(*seclevel-name*)

specifies the name of an installation-defined security level. This name corresponds to the number that is the minimum security level that a user must have to access the resource. The *seclevel-name* must be a member of the SECLEVEL profile in the SECDATA class.

When you specify SECLEVEL and the SECDATA class is active, RACF adds security level access checking to its other authorization checking. If global access checking does not grant access, RACF compares the security level allowed in the user profile with the security level required in the resource profile. If the security level in the user profile is less than the security level in the resource profile, RACF denies the access. If the security level in the user profile is equal to or greater than the security level in the resource profile, RACF continues with other authorization checking.

RACF does not perform security level checking for a started task that has the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF started

RALTER

procedures table or STARTED class. Also, RACF does not enforce security level information specified on profiles in the PROGRAM class.

If the SECDATA class is not active, RACF stores the name you specify in the resource profile. When the SECDATA class is activated and the name you specified is defined as a SECLEVEL profile, RACF can perform security level access checking for the resource profile. If the name you specify is not defined as a SECLEVEL profile, you are prompted to provide a valid *seclevel-name*.

NOSECLEVEL

specifies that the RALTER command is to delete the security level name from the profile. RACF no longer performs security level checking for the resource.

SESSION | NOSESSION

SESSION

controls the establishment of sessions between logical units under LU6.2. This operand is only valid for the APPCLU resource class. It allows the following suboperand to add, change, or delete SESSION segment field values when changing an APPCLU class profile.

CONVSEC | NOCONVSEC

CONVSEC(*security-checking-level*)

specifies the level or levels of security checking performed when conversations are established with the LU protected by this profile.

Security-checking-level can be one of the following levels.

Note: In general, you should select one of ALREADYV, AVPV, CONV, NONE or PERSISTV for each APPCLU profile.

NONE

All inbound allocate requests pass without RACF checking for a valid user ID. No RACROUTE REQUEST=VERIFY is issued.

CONV

APPC/MVS issues a RACROUTE REQUEST=VERIFY to verify the user ID and password for all inbound allocate requests.

ALREADYV

APPC/MVS RACF does *not* verify the user ID and password for any inbound allocate requests. If you specify ALREADYV, you assume that user IDs and passwords have already been verified by the partner LU. You must specify this only if the partner LU is trustworthy.

PERSISTV

Specifies persistent verification.

AVPV

The user ID/password is already verified *and* persistent verification is requested. In general, you should select one of NONE, CONV, and ALREADYV for each APPCLU profile.

NOCONVSEC

delete any existing conversation security parameters.

INTERVAL(*n*) | NOINTERVAL

INTERVAL(*n*)

sets the maximum number of days the session key is valid. This number, *n*, is in the range of 1 to 32767. If the key interval is longer than the installation maximum (set with SETROPTS SESSIONINTERVAL), the INTERVAL is not changed.

NOINTERVAL

There is no limit on the number of days the key is valid.

LOCK | NOLOCK

LOCK

marks the profile as locked.

NOLOCK

unlocks a previously locked profile.

SESSKEY | NOSESSKEY

SESSKEY(*session-key*)

changes the key for this profile. *Session key* can be expressed in two ways:

- x'y' where y is a 1-16-digit hexadecimal number
- z or 'z' where z is a 1-8 character string

If the entire 16 digits or 8 characters are not used, the field is padded to the right with binary zeros.

Note: Session keys are 64-bit Data Encryption Standard (DES) keys. With DES, 8 of the 64 bits are reserved for use as parity bits, so those 8 bits are not part of the 56-bit key. In hexadecimal notation, the DES parity bits are:X'0101 0101 0101 0101'. Any two 64-bit keys are equivalent DES keys if their only difference is in one or more of these parity bits. For instance, the following SESSKEY values, although appearing to be quite different, are equivalent because they differ only in the last bit of each byte:

- BDF0KM4Q, which is X'C2C4 C6F0 D2D4 F4D8'
- CEG1LN5R, which is X'C3C5 C7F1 D3D5 F5D9'

NOSESSKEY

deletes the session key for this profile

NOSESSION

deletes the SESSION segment from this profile.

SINGLEDSN | NOSINGLEDSN

SINGLEDSN

specifies that the tape volume can contain only one data set. SINGLEDSN is valid only for a TAPEVOL profile. If the volume already contains more than one data set, RACF issues a message and ignores the operand.

NOSINGLEDSN

specifies that the tape volume can contain multiple data sets, up to a maximum of 9999. NOSINGLEDSN is valid only for a TAPEVOL profile.

SSIGNON | NOSSIGNON

SSIGNON

defines the application key or a secured signon key and indicates the

RALTER

method you want to use to protect the key value within the RACF database. You can mask or encrypt the key. The *key-value* represents a 64-bit (8-byte) key that must be represented as 16 hexadecimal characters. The valid characters are 0 through 9 and A through F.

Notes:

1. As with RACF passwords, the database unload facility does not unload application keys or a secured signon keys.
2. The RLIST command does not list the value of the application key or the secured signon keys. Therefore, when you define the keys, you should note the value and keep it in a secure place.

KEYMASKED(*key-value*)

indicates that you want to mask the key value using the masking algorithm.

Notes:

1. You can specify this operand only once for each application key.
2. If you mask a key, you *cannot* encrypt it. These are mutually exclusive.

You can use the RLIST command described in “RLIST (List General Resource Profile)” on page 382 to be sure the key is protected.

KEYENCRYPTED(*key-value*)

indicates that you want to encrypt the key value.

Notes:

1. You can specify this operand only once for each application key.
2. If you encrypt a key, you *cannot* mask it. These are mutually exclusive.
3. A cryptographic product must be installed and active on the system.

You can use the RLIST command described in “RLIST (List General Resource Profile)” on page 382 to be sure the key is protected.

NOSSIGNON

specifies that the SSIGNON segment should be deleted.

STDATA | NOSTDATA

STDATA

used to control security for started tasks. STDATA should only be specified for profiles in the STARTED class.

USER(*userid* | =MEMBER) | NOUSER

USER(*userid*)

specifies the user ID to be associated with this entry.

RACF issues a warning message if the specified *userid* does not exist, but information is added to the STDATA segment. If the error is not corrected, RACF uses the started procedures table to process START requests that would have used this STARTED profile.

USER(=MEMBER)

specifies that the procedure name should be used as the user ID. If =MEMBER is specified for USER, a *group-name* value should be

specified for the GROUP operand. If =MEMBER is specified for both USER and GROUP, a warning message is issued and problems might result when the profile is used. For information, see *z/OS Security Server RACF Security Administrator's Guide*.

NOUSER

specifies the user ID should be deleted from this entry, leaving it unspecified. A warning message is issued because the absence of a user specification in the STDATA segment normally indicates that the segment information is incomplete. IF NOUSER is specified, RACF uses the started procedures table to process START requests that would have used this STARTED profile.

GROUP(group-name | =MEMBER) | NOGROUP**GROUP(group-name)**

specifies the group name to be associated with this entry.

RACF issues a warning message if the specified *group-name* does not exist. If *userid* and *group-name* are specified, RACF verifies that the user is connected to the group. If there is an error in the specification of the group name, the started task runs as an undefined user.

GROUP(=MEMBER)

specifies that the procedure name should be used as the group name. If =MEMBER is specified for GROUP, a *userid* value must be specified for the USER operand or RACF uses the started procedures table to assign an identifier for this started task. If =MEMBER is specified for both USER and GROUP, a warning message is issued and problems might result when the profile is used. For information, see *z/OS Security Server RACF Security Administrator's Guide*.

NOGROUP

specifies the group name should be deleted from this entry, leaving it unspecified. IF NOGROUP is specified, the started task runs with the default group of the specified user ID.

PRIVILEGED(NO | YES) | NOPRIVILEGED

specifies whether the started task should run with the RACF PRIVILEGED attribute. The PRIVILEGED attribute allows the started task to pass most authorization checking. No installation exits are called, no SMF records are generated, and no statistics are updated. (Note that bypassing authorization checking includes bypassing the checks for security classification of users and data.) For more information, see *z/OS Security Server RACF System Programmer's Guide*.

PRIVILEGED(NO) and NOPRIVILEGED indicate that the started task should run without the PRIVILEGED attribute.

TRACE(NO | YES) | NOTRACE

specifies whether a message should be issued to the operator when this entry is used to assign an ID to the started task.

If TRACE(YES) is specified, RACF issues an informational message to the operator to record the use of this entry when it is used to assign an ID to a started task. This record can be useful in finding started tasks

RALTER

that do not have a specific entry defined and in diagnosing problems with the user IDs assigned for started tasks.

TRACE(NO) and NOTRACE specify that an informational message should not be issued when this entry is used to assign an ID to the started task.

TRUSTED(NO | YES) | NOTRUSTED

specifies whether the started task should run with the RACF TRUSTED attribute. The TRUSTED attribute is similar to the PRIVILEGED attribute except that auditing can be requested using the SETROPTS LOGOPTIONS command. For more information about the TRUSTED attribute, see *z/OS Security Server RACF System Programmer's Guide*.

TRUSTED(NO) and NOTRUSTED indicate that the started task should run without the RACF TRUSTED attribute.

NOSTDATA

specifies that all the STDATA information for this entry should be deleted. When this entry is used, and no STDATA was specified (or when the STDATA has been deleted), then RACF issues a message and use the started procedures table to assign information for this START command.

SVFMR | NOSVFMR

SVFMR

defines profiles associated with a particular SystemView® for MVS application.

SCRIPTNAME(*script-name*) | NOSCRIPTNAME

SCRIPTNAME(*script-name*)

specifies the name of the list of default logon scripts associated with this application. This operand is optional. If you omit this operand, no scripts are changed for the application.

The *script-name* is the 1-8 character alphanumeric name of a member of an MVS partitioned data set (PDS). RACF accepts both upper- and lowercase characters for *script-name*, but lowercase characters are translated to uppercase.

The PDS member specified by *script-name* contains a list of other PDS members that contain the scripts associated with this application's profile. The PDS and members, including the member that contains the list of other members, are created by the administrator of the SystemView for MVS application.

NOSCRIPTNAME

specifies that the logon script name should be deleted from this entry.

PARMNAME(*parm-name*) | NOPARMNAME

PARMNAME(*parm-name*)

specifies the name of the parameter list associated with this application. This operand is optional. If this operand is omitted, no parameters are changed for the application.

The *parm-name* is the 1-8 character alphanumeric name of a member of an MVS partitioned data set (PDS). RACF accepts both upper- and lowercase characters for *parm-name*, but lowercase characters are translated to uppercase.

The PDS member specified by *parm-name* contains a list of other PDS members that contain the parameters associated with this application's profile. The PDS and members, including the list of other members, are created by System View for the MVS administrator.

NOPARMNAME

specifies that the parameter list name should be deleted from this entry.

NOSVFMR

specifies that the SVFMR segment should be deleted.

TIMEZONE | NOTIMEZONE

TIMEZONE({EIW} hh[.mm])

specifies the time zone in which a terminal resides. TIMEZONE is valid only for resources in the TERMINAL class; RACF ignores it for all other resources.

Specify TIMEZONE only when the terminal is not in the same time zone as the processor on which RACF is running. In this situation, TIMEZONE provides the information RACF needs to calculate the time and day values correctly. If you identify more than one terminal in the *profile-name* operand, all the terminals must be in the same time zone.

On TIMEZONE, you specify whether the terminal is east (E) or west (W) of the system and by how many hours (hh) and, optionally, minutes (mm). The terminal time zone is different from the processor time zone. Valid hour values are 0 through 11, and valid minute values are 00 through 59.

For example, if the processor is in New York and the terminal is in Los Angeles, specify TIMEZONE(W 3). If the processor is in Houston and the terminal is in New York, specify TIMEZONE(E 1).

If you change the local time on the processor (to accommodate daylight savings time, for instance), RACF adjusts its time calculations accordingly. If, however, the processor time zone and the terminal time zone do not change in the same way, you must adjust the terminal time zones yourself, as described for the WHEN(TIME) operand.

NOTIMEZONE

specifies that the terminal is in the same time zone as the processor. NOTIMEZONE is valid only for resources in the terminal class; RACF ignores it for all other resources.

TME | NOTME

TME

specifies that information for the Tivoli Security Management Application is to be added, changed, or deleted.

Note: The TME segment fields are intended to be updated only by the Tivoli Security Management Application, which manages updates, permissions, and cross references. A security administrator should only directly update Tivoli Security Management fields on an exception basis.

All TME suboperands, with the exception of those for ROLES, can be specified when changing a resource profile in the ROLE class. Conversely, only the ROLES suboperand can be specified when changing a resource profile in any other class.

CHILDREN | NOCHILDREN | ADDCHILDREN | DELCHILDREN

RALTER

CHILDREN(*profile-name* ...)

specifies the complete list of roles which inherit attributes from this role. A role is a discrete general resource profile defined in the ROLE class.

ADDCHILDREN(*profile-name* ...)

specifies the addition of specific child roles to the current list of roles.

DELCHILDREN(*profile-name* ...)

specifies the removal of specific child roles from the current list of roles.

NOCHILDREN

specifies the removal of the entire list of child roles.

GROUPS | NOGROUPS | ADDGROUPS | DELGROUPS

GROUPS(*group-name* ...)

specifies the complete list of groups which should be permitted to resources defined in this role profile.

Group-name should be the name of a defined group.

ADDGROUPS(*group-name* ...)

specifies the addition of specific groups to the current list of groups.

DELGROUPS(*group-name* ...)

specifies the removal of specific groups from the current list of groups.

NOGROUPS

specifies the removal of the entire list of groups.

PARENT | NOPARENT

PARENT(*profile-name*)

specifies the name of a role from which this role inherits attributes. A role is a discrete general resource profile defined in the ROLE class.

NOPARENT

specifies the deletion of the parent role from this profile.

RESOURCE | NORESOURCE | ADDRESOURCE | DELRESOURCE

RESOURCE(*resource-access-specification* ...)

specifies the complete list of resources and associated access levels for groups defined in this role profile.

One or more *resource-access-specification* values can be specified, each separated by blanks. Each value should contain no imbedded blanks and should have the following format:

origin-role: class-name: profile-name: access-authority
[: *conditional-class: conditional-profile*]

where *origin-role* is the name of the role profile from which the resource access is inherited. *Class-name* is an existing resource class name and *profile-name* is a resource profile defined in that class. The *access-authority* is the authority (NONE, EXECUTE, READ, UPDATE, CONTROL, or ALTER) with which groups in the role definition should be permitted to the resource.

The *conditional-class* is a class name (APPCPORT, CONSOLE, JESINPUT, PROGRAM, TERMINAL, or SYSID) for conditional access permission, and is followed by the *conditional-profile* value, a resource profile defined in the conditional class.

ADDRESOURCE(*resource-access-specification ...*)

specifies the addition of specific resource-access-specifications to the current list.

DELRESOURCE(*resource-access-specification ...*)

specifies the removal of specific resource-access-specifications from the current list.

NORESOURCE

specifies the removal of the entire list of resources.

ROLES | NOROLES | ADDROLES | DELROLES

ROLES(*role-access-specification ...*)

specifies a list of roles and associated access levels related to this profile.

One or more *role-access-specification* values can be specified, each separated by blanks. Each value should contain no imbedded blanks and should have the following format:

role-name:access-authority[:conditional-class:conditional-profile]

where *role-name* is a discrete general resource profile defined in the ROLE class. The *access-authority* is the authority (NONE, EXECUTE, READ, UPDATE, CONTROL, or ALTER) with which groups in the role definition should be permitted to the resource.

The *conditional-class* is a class name (APPCPORT, CONSOLE, JESINPUT, PROGRAM, TERMINAL, or SYSID) for conditional access permission, and is followed by the *conditional-profile* value, a resource profile defined in the conditional class.

ADDROLES(*role-access-specification ...*)

specifies that specific roles and access levels are to be added to the current list.

DELROLES(*role-access-specification ...*)

specifies that specific roles from the current list of roles are to be removed.

NOROLES

specifies that the entire list of roles be removed.

NOTME

specifies that RACF delete the TME segment from the profile.

TVTOC | NOTVTOC

TVTOC

specifies, for a TAPEVOL profile, that RACF is to create a TVTOC in the TAPEVOL profile when a user creates the first output data set on the volume.

Specifying TVTOC affects the access list for the TAPEVOL profile:

1. When RACF processes the RALTER command with the TVTOC operand, it places the user ID of the command issuer (perhaps the tape librarian) in the access list with ALTER authority.

RALTER

2. When the first output data set is created on the volume, RACF adds the user ID associated with the job or task to the access list with ALTER authority.

See *z/OS Security Server RACF Security Administrator's Guide* for further information.

The TVTOC operand is valid only for a discrete profile in the TAPEVOL class. If you specify TVTOC and the volume already contains a TVTOC, RACF issues a message and ignores the operand.

NOTVTOC

specifies that RACF cannot create a TVTOC in the resource profile. The NOTVTOC operand is valid only for a discrete profile in the TAPEVOL class. It is also invalid if a TVTOC with at least one entry already exists in the TAPEVOL profile. When NOTVTOC is invalid, RACF issues a message and ignores the operand. If your installation uses HSM and you activate tape data set protection, the TVTOC for HSM tapes might become too large. To avoid this problem, issue the following RALTER command:

```
RALTER TAPEVOL HSMHSM NOTVTOC
```

UACC(*access-authority*)

specifies the universal access authority to be associated with this resource. The universal access authorities are ALTER, CONTROL, UPDATE, READ, EXECUTE (for controlled programs only), and NONE.

Notes:

1. For tape volumes and DASD volumes, RACF treats CONTROL authority as UPDATE authority.
2. For all other resources listed in the class descriptor table, RACF treats CONTROL and UPDATE authority as READ authority.
3. If a user accessing a data set has the RESTRICTED attribute, RACF treats the universal access authority (UACC) as NONE for that access attempt.

WARNING | NOWARNING

WARNING

specifies that even if access authority is insufficient, RACF is to issue a warning message and allow access to the resource. RACF also records the access attempt in the SMF record if logging is specified in the profile. RACF does **not** issue a warning message for a resource when the resource is:

- In the PROGRAM class
- When the resource is in a class for which an application has built in-storage profiles using RACROUTE REQUEST=LIST

Some applications, such as IMS and CICS, load all the profiles for a given class into storage. After these profiles are in storage, the applications can do a “fast” authorization check using RACROUTE REQUEST=FASTAUTH. Fast authorization checking is different from normal authorization checking in several ways. One difference is that, in some cases, fast authorization checking does not issue warning messages, notification messages or support auditing. In cases where it does not, return and reason codes are returned to the application to allow support of these functions. The application can examine the return and reason codes and use RACROUTE REQUEST=AUTH to create the messages and audit records. If the application uses RACROUTE REQUEST=AUTH to support auditing, the specified user is notified. Otherwise, notification, warning, and so on does not occur.

For details on using RACF with IMS, see *z/OS Security Server RACF Security Administrator's Guide*. For details on using RACF with CICS, refer to *CICS RACF Security Guide*.

NOWARNING

specifies that if access authority is insufficient, RACF is to deny the user access to the resource and not issue a warning message.

WHEN([DAYS(*day-info*)] [TIME(*time-info*)])

specifies, for resources in the **TERMINAL** class, the days of the week or the hours in the day when the terminal can be used to access the system. The day-of-week and time restrictions apply only when a user logs on to the system; that is, RACF does not force the user off the system if the end-time occurs while the user is logged on.

If you specify the **WHEN** operand, you can restrict the use of the terminal to certain days of the week or to a certain time period on each day. You can also restrict access to both certain days of the week and to a certain time period within each day.

To allow use of the terminal only on certain days, specify **DAYS(*day-info*)**, where *day-info* can be anyone of the following:

ANYDAY

allows use of the terminal on any day.

WEEKDAYS

allows use of the terminal only on weekdays (Monday through Friday).

day...

allows use of the terminal only on the days specified, where *day* can be **MONDAY**, **TUESDAY**, **WEDNESDAY**, **THURSDAY**, **FRIDAY**, **SATURDAY**, or **SUNDAY**, and you can specify the days in any order.

To allow use of the terminal only during a certain time period of each day, specify **TIME(*time-info*)**, where *time-info* can be any one of the following:

ANYTIME

RACF allows use of the terminal at any time.

start-time:end-time

RACF allows use of the terminal only during the specified time period. The format of both *start-time* and *end-time* is *hhmm*, where *hh* is the hour in 24-hour notation (00 through 24) and *mm* is the minutes (00 through 59) within the range 0001–2400. Note that 2400 indicates 12:00 a.m. (midnight).

If *start-time* is greater than *end-time*, the interval spans midnight and extends into the following day.

Specifying *start-time* and *end-time* is straightforward when the processor on which RACF is running and the terminal are in the same time zone; you specify the time values in local time.

If, however, the terminal is in a different time zone from the processor and you want to restrict access to certain time periods, you have two choices. You can specify the **TIMEZONE** operand to allow RACF to calculate the time and day values correctly. Or, you can adjust the time values yourself, by translating the *start-time* and *end-time* for the terminal to the equivalent local time for the processor.

RALTER

For example, assume that the processor is in New York and the terminal is in Los Angeles, and you want to allow access to the terminal from 8:00 A.M. to 5:00 P.M. in Los Angeles. In this situation, you would specify TIME(1100:2000). If the processor is in Houston and the terminal is in New York, you would specify TIME(0900:1800).

If you omit DAYS and specify TIME, the time restriction applies to any day-of-week restriction already specified in the profile. If you omit TIME and specify DAYS, the days restriction applies to any time restriction already specified in the profile. If you specify both DAYS and TIME, RACF allows use of the terminal only during the specified time period and only on the specified days.

Examples

Table 53. RALTER Examples

Example 1	<i>Operation</i>	User TRA02 wants to change the owner and universal access for terminal TERMID01 and restrict use of the terminal to weekdays during regular business hours (8:00 A.M. to 6:00 P.M.).
	<i>Known</i>	User TRA02 has the SPECIAL attribute.
		Terminal TERMID01 is defined to RACF. Terminal TERMID01 is in the same time zone as the processor on which RACF is running.
	<i>Command</i>	User TRA02 wants to issue the command as a RACF TSO command. RALTER TERMINAL TERMID01 OWNER(TRA02) UACC(ALTER) WHEN(DAYS(WEEKDAYS)TIME(0800:1800))
Example 2	<i>Defaults</i>	None
	<i>Operation</i>	User RFF23 wants to delete the two data fields associated with the terminal T3E8. The user wants to be notified whenever the terminal profile denies access to the terminal.
	<i>Known</i>	User RFF23, who is a RACF-defined user, is the owner of the T3E8 terminal entry.
	<i>Command</i>	User RFF23 wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @. @RALTER TERMINAL T3E8 NODATA NOAPPLDATA NOTIFY(RFF23)
Example 3	<i>Defaults</i>	None
	<i>Operation</i>	User ADM1 wants to delete the data fields associated with the generic profile * in the TERMINAL class.
	<i>Known</i>	User ADM1 has the SPECIAL attribute.
	<i>Command</i>	User ADM1 wants to issue the command as a RACF TSO command. RALTER TERMINAL * NODATA NOAPPLDATA
Example 4	<i>Defaults</i>	None
	<i>Operation</i>	User PAYADM1 wants to add the PAYROLL category to the list of security categories known to RACF.
	<i>Known</i>	User PAYADM1 has the SPECIAL attribute. RACF security category checking is active.
	<i>Command</i>	User PAYADM1 wants to issue the command as a RACF TSO command. RALTER SECDATA CATEGORY ADDMEM(PAYROLL)
	<i>Defaults</i>	None

Table 53. RALTER Examples (continued)

Example 5	<i>Operation</i>	User RFF22 wants to add volume TAP02 to the tape volume set, change the level of the tape volume set, and change the AUDIT and GLOBALAUDIT logging options.
	<i>Known</i>	User RFF22 is the owner of the tape volume set.
		User RFF22 has the AUDITOR attribute.
		TAP01 is a volume of the tape volume set.
Example 6		User RFF22 wants to issue the command as a RACF TSO command.
	<i>Command</i>	RALTER TAPEVOL TAP01 AUDIT(SUCCESS(READ)) LEVEL(22) GLOBALAUDIT(SUCCESS(UPDATE)FAILURES(READ)) ADDVOL(TAP02)
	<i>Defaults</i>	None
	<i>Operation</i>	User ADM1 wants to add AMASPZAP to the in-storage profile table of controlled programs. AMASPZAP requires program-accessed data set checking.
Example 7	<i>Known</i>	User ADM1 has the SPECIAL attribute. AMASPZAP resides in SYS1.LINKLIB on the SYSRES volume. RACF program control is active.
		User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	RALTER PROGRAM AMASPZAP ADDMEM('SYS1.LINKLIB'/SYSRES/PADCHK)
	<i>Defaults</i>	None
Example 8	<i>Operation</i>	User ADM1 wants to add all load modules that start with IKF to the in-storage profile table of controlled programs. These load modules do not require program-accessed data set checking. User ADM1 wants to direct the command to run at the local node under the authority of user EMILIE and prohibit the command from being automatically directed to other nodes.
	<i>Known</i>	Users ADM1 and EMILIE have the SPECIAL attribute. All load modules whose names begin with IKF reside in SYS1.COBLIB on the SYSRES volume. RACF program control is active. Users ADM1 and EMILIE have an already established user ID association.
		User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	RALTER PROGRAM IKF* ADDMEM('SYS1.COBLIB'/SYSRES/NOPADCHK) ONLYAT(.EMILIE)
Example 9	<i>Results</i>	The command is only processed on the local node and not automatically directed to any other nodes in the RRSF configuration.
	<i>Operation</i>	The security administrator wants to change the key value of a profile in the PTKTDATA class so the value becomes encrypted.
	<i>Known</i>	NONNEL is the user ID of the security administrator.
		The profile name is TSOR004.
Example 9		The <i>key-value</i> is B004194019641980.
		The security administrator wants to issue the command as a RACF TSO command.
	<i>Command</i>	RALTER PTKTDATA TSOR004 SSIGNON(KEYENCRYPTED(B004194019641980))
	<i>Defaults</i>	None
Example 9	<i>Operation</i>	The administrator wants to change the script and parameter definitions for an existing SystemView for MVS application that has been defined to the SYSMVIEW class.
	<i>Known</i>	The new script definition is APPL2SC.
		The new parameter definition is APPL2P.
	<i>Command</i>	RALTER SYSMVIEW APPL1.HOST1.USER1 SVFMR(SCRIPTNAME(APPL2SC) PARMNAME(APPL2P))
Example 9	<i>Defaults</i>	None

RALTER

Table 53. RALTER Examples (continued)

Example 10	<i>Operation</i>	Local realm KRB2000.IBM.COM is being defined with a minimum ticket lifetime of 5 minutes, a default ticket lifetime of 10 hours, a maximum ticket lifetime of 24 hours, and a password of 744275. All of the ticket lifetime values are specified in seconds.
	<i>Known</i>	The administrator has access to the KERBDFLT profile in the REALM class.
	<i>Command</i>	RALTER REALM KERBDFLT KERB(KERBNAME(KRB2000.IBM.COM) MINTKTLFE(300) + DEFTKTLFE(36000) MAXTKTLFE(86400) PASSWORD(744275))
Example 11	<i>Defaults</i>	None
	<i>Operation</i>	A trust relationship is being defined between the kerb390.endicott.ibm.com realm and the realm at ker2000.endicott.ibm.com.
	<i>Known</i>	The administrator has access to the /.../KERB390.ENDICOTT.IBM.COM/KRBTGT/KER2000.ENDICOTT.IBM.COM profile in the REALM class.
	<i>Command</i>	RALTER REALM /.../KERB390.ENDICOTT.IBM.COM/KRBTGT/KER2000.ENDICOTT.IBM.COM KERB(PASSWORD(12345678))
Example 12	<i>Defaults</i>	None
	<i>Operation</i>	The system default EIM values are being altered by changing the DOMAINDN and disabling it.
	<i>Known</i>	IRR.PROXY.DEFAULTS is the profile being changed in the FACILITY class. The EIM domain distinguished name begins with Pok EIM Domain,o=IBM,c=US.
	<i>Command</i>	RALTER FACILITY IRR.PROXY.DEFAULTS EIM(DOMAINDN('ibm-eimDomainName=Pok EIM Domain,o=IBM,c=US') OPTIONS(DISABLE))
	<i>Defaults</i>	None

RDEFINE (Define General Resource Profile)

Purpose

Use the RDEFINE command to define to RACF all resources belonging to classes specified in the class descriptor table. You can also use the RDEFINE command to create entries in the global access checking table and in the lists of security categories and security levels, and to define classes (as profiles in the RACGLIST class) for which RACF saves RACLISTed results on the RACF database.

The RDEFINE command adds a profile for the resource to the RACF database in order to control access to the resource. It also places your user ID on the access list and gives you ALTER authority to the resource unless SETROPTS NOADDCREATOR is in effect.

You cannot use the RDEFINE command to define users, groups, data sets, certificates, certificate key rings, or certificate mappings.

To have changes take effect after defining a generic profile if the class is not RACLISTed by either the SETROPTS RACLIST or RACROUTE REQUEST=LIST, GLOBAL=YES, one of the following steps is required:

- The security administrator issues the SETROPTS command:

```
SETROPTS GENERIC(class-name) REFRESH
```

See the SETROPTS command for authorization requirements.

- The user of the resource logs off and logs on again.

To have changes take effect after defining a generic profile if the class is RACLISTed, the security administrator issues the following command:

```
SETROPTS RACLIST(class-name) REFRESH
```

For more information, refer to *z/OS Security Server RACF Security Administrator's Guide*.

Attention:

- When the RDEFINE command is issued from ISPF, the TSO command buffer (including SESSKEY and SSIGNON) is written to the ISPLOG data set. As a result, you should not issue this command from ISPF or you must control the ISPLOG data set carefully.
- If the RDEFINE command is issued as a RACF operator command, the command and all data is written to the system log. Therefore, use of RDEFINE as a RACF operator command should either be controlled or you should issue the command as a TSO command.

Issuing Options

The following table identifies the eligible options for issuing the RDEFINE command:

Table 54. How the RDEFINE Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	Yes	Yes

RDEFINE

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To create a group profile, see “ADDGROUP (Add Group Profile)” on page 24.
- To create a data set profile, see “ADDSD (Add Data Set Profile)” on page 33.
- To create a user profile, see “ADDUSER (Add User Profile)” on page 48.
- To permit or deny access to a general resource profile, see “PERMIT (Maintain Resource Access Lists)” on page 247.
- To change a general resource profile, see “RALTER (Alter General Resource Profile)” on page 303.
- To delete a general resource profile, see “RDELETE (Delete General Resource Profile)” on page 371.
- To obtain a list of general resource profiles, see “SEARCH (Search RACF Database)” on page 408.
- To list a general resource profile, see “RLIST (List General Resource Profile)” on page 382.

Authorization Required

When issuing the RDEFINE command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

To use the RDEFINE command, you must have the SPECIAL attribute or be authorized as follows:

- If you have CLAUTH authority for the GLOBAL class, and group-SPECIAL authority in a group, you can add members whose high-level qualifier is the group name or a user ID owned by the group. This applies only to classes that are sensitive to high-level qualifiers, such as JESINPUT and DATASET.
- If the resource to be defined is not already defined to RACF as a member of a resource group, you must be authorized to define resources for the specified class. (This authority can be established with the CLAUTH operand on the ADDUSER or ALTUSER command.)
- If the resource to be defined is a discrete name already defined to RACF as a member of a resource group, you can define it as a resource to RACF if you have ALTER authority, or if the resource group profile is within the scope of a group in which you have the group-SPECIAL attribute, or if you are the owner of the resource group profile. If authority conflicts arise because the resource is a member of more than one group and the user's authority in those groups differs, RACF resolves the conflict by using the least restrictive authority (unless modified by the installation).
- If you define a profile in the FILE or DIRECTRY class, one of the following must be true:
 - The second qualifier of the profile name must match your user ID
 - You must have the SPECIAL attribute

- The profile name must be within the scope of a group in which you have the group-SPECIAL attribute
- If the SETROPTS GENERICOWNER option is in effect, and if a generic profile already exists, *more specific* profiles that protect the same resources can be created only by the following users:
 - The owner of the existing generic profile
 - A user with group-SPECIAL if the group owns the profile
 - A user with group-SPECIAL if the owner of the existing profile is in a group within the scope of the command issuer's group-SPECIAL attribute
 - A user with the SPECIAL attribute

Notes:

1. GENERICOWNER does not apply to the PROGRAM general resource class.
 2. For additional information on the GENERICOWNER option and restricting the creation of general resource profiles, see *z/OS Security Server RACF Security Administrator's Guide*.
- To assign a security category to a profile, you must have the category in your user profile.
 - To assign a security level to a profile, your own profile must have a security level that is equal to or greater than the security level you are defining.
 - To use the ADDMEM operand, see the description of the ADDMEM operand for information on the authority required to use the operand.
 - To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).
 - To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.
 - To define segments other than the base segment, such as DLFDATA, you must have SPECIAL or your installation must permit you to do so through field-level access checking.
 - To assign a security label to a profile, you must have READ access to the security label profile. However, the security administrator can limit the ability to assign security labels to only users with the SPECIAL attribute.

Model Profiles: To specify a model profile (using, as required, FROM, FCLASS, FGENERIC, and FVOLUME), you must have sufficient authority over the model profile—the “from” profile. RACF makes the following checks until one of the conditions is met:

- You have the SPECIAL attribute.
- The “from” profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You are the owner of the “from” profile.
- If the FCLASS operand is DATASET, the high-level qualifier of the profile name (or the qualifier supplied by the naming conventions routine or a command installation exit) is your user ID.

For discrete profiles only:

- You are on the access list in the “from” profile with ALTER authority. (If you have any lower level of authority, you cannot use the profile as a model.)

RDEFINE

- Your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is in the access list in the “from” profile with ALTER authority. (If any group that RACF checked has any lower level of authority, you cannot use the profile as a model.)
- The universal access authority (UACC) is ALTER.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the RDEFINE command is:

```
[subsystem-prefix]{RDEFINE | RDEF}

    class-name
    (profile-name-1 ...)
    [ ADDCATEGORY(category-name ...) ]
    [ ADDMEM(member ...) ]
    [ APPLDATA('application-data') ]
    [ AT([node].userid ...) | ONLYAT([node].userid ...) ]
    [ AUDIT( access-attempt [(audit-access-level)] ...) ]
    [ DATA('installation-defined-data') ]
    [ DLFDATA(
        [ RETAIN( YES | NO ) ]
        [ JOBNAMES(jobname-1 ...) ]
    ) ]
    [ EIM(
        [DOMAINDN(eim_domain_dn)]
        [OPTIONS (ENABLE | DISABLE)]
        [LOCALREGISTRY(registry_name)] ) ]
    [ FCLASS(profile-name-2-class) ]
    [ FGENERIC ]
    [ FROM(profile-name-2) ]
    [ FVOLUME(profile-name-2-serial) ]
    [ KERB (
        [ DEFTKTLFE(def-ticket-life) ]
        [ ENCRYPT (
            [ DES | NODES ]
            [ DES3 | NODES3 ]
            [ DESD | NODESD ]
        ) ]
        [ KERBNAME(kerberos-realm-name) ]
        [ MAXTKTLFE(max-ticket-life) ]
        [ MINTKTLFE(min-ticket-life) ]
        [ PASSWORD(kerberos-password) ] ]
    [ LEVEL(nn) ]
    [ NOTIFY(userid) ]
    [ OWNER (userid or group-name) ]
    [PROXY(
        [ LDAPHOST(ldap_url)]
        [ BINDDN(bind_distinguished_name)]
        [ BINDPW(bind_password)] ) ]
    [ SECLABEL(seclabel-name) ]
    [ SECLEVEL(seclabel-name) ]
```

```

[ SESSION(
  [ CONVSEC( NONE | CONV | PERSISTV
             | ALREADYV | AVPV )
  ]
  [ INTERVAL(n) ]
  [ LOCK ]
  [ SESSKEY(session-key) ]
  ) ]
[ SINGLED SN ]
[ SSIGNON(
  [ KEYMASKED(key-value)
  | KEYENCRYPTED(key-value)] ]
[ STDATA(
  [ USER(userid | =MEMBER) ]
  [ GROUP(group-name | =MEMBER) ]
  [ PRIVILEGED( NO | YES) ]
  [ TRACE( NO | YES) ]
  [ TRUSTED( NO | YES) ]
  ) ]
[ SVFMR(
  [ SCRIPTNAME(script-name) ]
  [ PARMNAME(parm-name) ]
  ) ]
[ TIMEZONE( {E | W} hh[.mm] ) ]
[ TME(
  [ CHILDREN(profile-name ...) ]
  [ GROUPS(group-name ...) ]
  [ PARENT(profile-name) ]
  [ RESOURCE(resource-access-specification ...) ]
  [ ROLES(role-access-specification ...) ]
  ) ]
[ TVTOC ]
[ UACC(access-authority) ]
[ WARNING ]
[ WHEN( [DAYS(day-info)] [TIME(time-info)] ) ]

```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

class-name

specifies the name of the class to which the resource belongs. The valid class

RDEFINE

names are those defined in the class descriptor table. For a list of general resource classes supplied by IBM, see Appendix B, "Description of RACF Classes" on page 507.

This operand is required and must be the first operand following RDEFINE.

This command is not intended to be used for profiles in the following classes:

- DCEUUIDS
- DIGTCERT
- DIGTNMAP
- DIGTRING
- NDSLINK
- NOTELINK
- ROLE
- UNIXMAP

Note: If you have the CLAUTH attribute (class authority) to a member or grouping class, the member or grouping class must be active in order for you to define profiles in that class.

profile-name-1

specifies the name of the discrete or generic profile you want to add to the specified class. RACF uses the class descriptor table to determine if the class is defined to RACF, the syntax of resource names within the class, and whether the resource is a group resource. For more information, see Appendix A, "Naming Considerations for Resource Profiles" and *z/OS Security Server RACF Security Administrator's Guide*.

Mixed case profile names are accepted and preserved when *class-name* refers to a class defined in the class descriptor table with CASE=ASIS.

This operand is required and must be the second operand following RDEFINE.

- If you specify more than one profile name, you must enclose the list of names in parentheses.
- In general, you should not specify profile names within single quotes because most classes will not allow this, and the RDEFINE command will fail. Classes such as FACILITY (or others whose class definition allows "any" character as the first character) will allow RDEFINE to work, but this will result in defining a profile whose name contains the single quote. As a result, during authorization checking, the profile might not protect the resources intended to be protected. In fact, such a profile can only work if you also have a resource manager that encloses its resource names in single quotes, but most resource managers do not.
- If you specify *class-name* as GLOBAL, *profile-name-1* must be either DATASET or a valid class name (other than a resource group class) as specified in the class descriptor table. If you specify *class-name* as GLOBAL or SECDATA and also specify ADDMEM or DELMEM, you can specify only one profile name.
- If you want RACF to store the results from a SETROPTS RACLIST or a RACROUTE REQUEST=LIST,GLOBAL=YES in the RACGLIST class, you must define the base profile for the class by issuing the command:

```
RDEFINE RACGLIST profile-name-1
```

where *profile-name-1* is a valid class in the class descriptor table. If the RACGLIST class is active when the class *profile-name-1* is RACLISTed,

RACF stores the RACLISTed results as profile-name-1_nnnnn profiles on the RACF database. For example, the command:

```
RDEFINE RACGLIST DASDVOL
```

creates a base profile DASDVOL. A subsequent

```
SETROPTS RACLIST(DASDVOL)
```

stores the RACLIST results as profiles DASDVOL_00001, DASDVOL_00002, and so on, in the RACGLIST class.

GLOBAL, RACGLIST, USER, CONNECT, GROUP, and DATASET, as well as classes that are marked in the class list in Appendix B, “Description of RACF Classes” on page 507, as “Profiles are not allowed in the class” like DIRAUTH, and classes that are marked as “Not for use on RACF commands” like SCDMBR, are not allowed to be specified as profile names for RACGLIST. The only grouping classes allowed to be specified with RACGLIST are NODES and RACFVARS.

- If *class-name* is a resource grouping class (other than NODES or RACFVARS), you cannot specify a generic *profile-name-1*. If *class-name* is DLFCLASS, you should not specify a generic *profile-name-1* as it is ignored by DLF processing.
- If you specify *class-name* as PROGRAM, you can specify only one profile name, and you must specify the ADDMEM or DELMEM operand.
- If you specify *class-name* as PROGRAM, *profile-name* must be the name of a load module. If you specify the full name of the load module, the profile applies only to that module. If you specify the last character of the name as an asterisk (*), the profile applies to all load modules that match the preceding part of the name, and these load modules must all reside in the same library. For example, IKF* identifies all load module names that begin with IKF. If you specify *profile-name* as an asterisk (*), the profile applies to all load modules that reside in the library you identify on the ADDMEM or DELMEM operand.
- If you are activating field-level access checking, you must specify *class-name* as FIELD. To define a profile (*profile-name-1*) in the FIELD class, you must follow the naming conventions described in the section on field-level access checking in *z/OS Security Server RACF Security Administrator's Guide*.
- If you specify *class-name* as STARTED, you must specify two qualifiers for the profile name. For more information on security for the STARTED class, see *z/OS Security Server RACF Security Administrator's Guide*.

Notes:

1. Do not specify a generic character unless SETROPTS GENERIC (or SETROPTS GENCMD) is in effect.
2. RACF processes each resource you specify independently, and all operands you specify apply to each named resource. If an error occurs while it is processing a resource, RACF issues a message and continues processing with the next resource.

ADDCATEGORY(*category-name* ...)

specifies one or more names of installation-defined security categories. The names you specify must be defined as members of the CATEGORY profile in the SECDATA class. (For information on defining security categories, see *z/OS Security Server RACF Security Administrator's Guide*.)

RDEFINE

When the SECDATA class is active and you specify ADDCATEGORY, RACF performs security category checking in addition to its other authorization checking. If a user requests access to a resource, RACF compares the list of security categories in the user's profile with the list of security categories in the resource profile. If RACF finds any security category in the resource profile that is not in the user's profile, RACF denies access to the resource. If the user's profile contains all the required security categories, RACF continues with other authorization checking.

Note: RACF does not perform security category checking for a started task with the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF started procedures table or STARTED class. Also, RACF does not enforce security category information specified on profiles in the PROGRAM class.

ADDMEM(*member ...*)

specifies the member names that RACF is to add to the profile indicated by *profile-name-1*. The meaning of "member" varies, depending on the class.

You can use the ADDMEM operand to perform tasks such as defining security categories and security levels, entries in the global access checking table, and entries for program control as described in the following sections.

Mixed case member names are accepted and preserved when *class-name* refers to a class defined in the class descriptor table with CASE=ASIS. When *class-name* is GLOBAL and *profile-name* is the name of a class defined in the class descriptor table with CASE=ASIS, the name part of a member entry in the GLOBAL access table is preserved as entered.

If you define a profile and use generic characters such as (*) to add members to the profile, RLIST RESGROUP will not return any of the matching profiles in its output because it does not support generic matches. For example, you have:

```
RDEF GIMS GIMSGRP ADDMEM(ABC*)
```

and you are looking for a specific member, so you enter:

```
RLIST TIMS ABCD RESGROUP
```

The GIMS profile GIMSGRP will not appear in the output.

Note: When considering this example, if you are unable to define the profile ABCD, it might be due to a generic definition somewhere in GIMS.

When specifying the &RACUID keyword with ADDMEM, generic characters such as the asterisk (*) and the percent sign (%) cannot follow the keyword.

In addition to the authority needed to issue the RDEFINE command, you need one of the following authorities to add members using the RDEFINE command:

1. For classes other than PROGRAM, SECDATA, GLOBAL, RACFVARS, and NODES, if the member resources are already RACF-protected by a member class profile or as a member of a profile in the same grouping class, one of the following must be true:
 - You have ALTER access authority to the member.
 - You are the owner of the member resource.
 - The member resource is within the scope of a group in which you have the group-SPECIAL attribute.

- You have the SPECIAL attribute.
2. For classes other than PROGRAM, SECADATA, GLOBAL, RACFVARS, and NODES, if the member resources are not RACF-protected (that is, there is no profile defined for that member), one of the following must be true:
 - You have CLAUTH authority to define resources in the member resource class.
 - You have the SPECIAL attribute.
 3. To add a member to a profile in the RACFVARS or NODES class, one of the following must be true:
 - You have CLAUTH authority to define resources in the specified class (for example, RACFVARS or NODES).
 - You have the SPECIAL attribute.
 - You are the owner of the profile indicated by *profile-name-1*.
 - You have ALTER access authority to the profile indicated by *profile-name-1*.
 4. To add a member to a profile in the PROGRAM or SECADATA class, one of the following must be true:
 - You have CLAUTH authority to define resources in the specified class (for example, PROGRAM or SECADATA).
 - You have the SPECIAL attribute.
 5. To add a member to a profile in the GLOBAL class (other than the GLOBAL DATASET, GLOBAL DIRECTORY, or GLOBAL FILE profile) where the syntax is:

```
RDEF GLOBAL class-name ADDMEM(resource-name/access-level)
```

one of the following must be true:

- If the profile *resource-name* is already RACF-protected by a profile in class *class-name*:
 - You have ALTER access authority to the profile *resource-name* in class *class-name*.
 - You are the OWNER of the profile *resource-name*.
 - The profile *resource-name* in class *class-name* is within the scope of a group in which you have the group-SPECIAL attribute.
 - You have the SPECIAL attribute.
6. If the profile *resource-name* is not already RACF-protected (that is, there is no profile defined for that member in class *class-name*):
 - You have CLAUTH authority to define resources in the class *class-name*.
 - You have the SPECIAL attribute.
 7. To add a member to the GLOBAL DATASET profile, one of the following must be true:
 - You are the owner of the DATASET profile in the GLOBAL class.
 - The member is within the scope of a group in which you have the group-SPECIAL attribute, or the high-level qualifier of the member name is your userid.
 - You have the SPECIAL attribute.
 8. To add a member to the GLOBAL DIRECTORY or GLOBAL FILE profile, you must have the SPECIAL attribute.

RACF ignores the ADDMEM operand if the class name you specify is not a resource grouping class, GLOBAL, SECADATA, NODES, or PROGRAM.

RDEFINE

Specifying member on the ADDMEM operand

The following sections describe how to specify members for each of the following classes:

- Resource Grouping Class
- GLOBAL
- SECDATA
- NODES
- PROGRAM

The descriptions for these classes are below.

- **When a Resource Grouping Class is the class-name**

Resource Grouping Class: If the class-name is a resource grouping class, the members you specify through the ADDMEM operand protects the resources in the related member class.

If generic profile checking is active for the related member class, you can include a generic character (*, **, &, or % only) in the member to protect multiple resources.

For more information on resource grouping classes and their related member classes, see *z/OS Security Server RACF Security Administrator's Guide*.

- **When GLOBAL is the class-name**

Global Access Checking: You can define an entry in the global access checking table by issuing the RDEFINE command with the following operands:

- GLOBAL as the class-name
- The appropriate resource class name as profile-name
- ADDMEM with the name of the entry you are defining (as *member*). (If the name you specify as *member* contains a generic character (* or %), generic profile checking (SETROPTS command with the GENERIC operand) must be active for the resource class you specify as *profile-name*.)
- The access level you are assigning to the entry (member) using the following format:

member[/{ALTER|CONTROL|NONE|READ|UPDATE}]

The format of this command is as follows:

```
RDEFINE GLOBAL profile-name ADDMEM(member/access-level)
```

Each entry you define controls global access checking for the resources matching that entry name.

Attention

Because RACF performs global access checking before security classification processing, global access checking might allow access to a resource you are protecting with a security category, security level, or both. To avoid a security exposure to a sensitive resource, do not define an entry in the global access checking table for a resource you are protecting with security classification processing.

When you define an entry in the global access checking table, specify *member* on the ADDMEM operand as described in the following sections.

– **Global Access Checking for Data Sets**

When you define an entry in the global access checking table for a data set, enclose the entry name in quotes if you do not want your TSO prefix (which might be your user ID) used as the high-level qualifier of the entry name.

For example, assume that your user ID is SMITH. If you issue the following command:

```
RDEFINE GLOBAL DATASET ADDMEM('SMITH.ABC'/READ)
```

you define the entry SMITH.ABC in the global access table.

If you do not enclose the entry name in quotes, your TSO prefix is used as the high-level qualifier of the entry name. For example, if you issue the following command:

```
RDEFINE GLOBAL DATASET ADDMEM(ABC/READ)
```

you define the entry SMITH.ABC in the global access table.

If the entry name you specify contains * as the high-level qualifier and you do not enclose the name in quotes, RACF creates the entry exactly as you specify it (your TSO prefix is **not** used as the high-level qualifier of the entry name). For example, if you issue the following command:

```
RDEFINE GLOBAL DATASET ADDMEM(*.ABC/READ)
```

you define the entry *.ABC in the global access table. If you enclose *.ABC in quotes, you define the same entry (*.ABC) in the global access table.

– **Global Access Checking for General Resources**

To define an entry in the global access checking table for a general resource, specify any valid class name as a profile name in the class descriptor table supplied by IBM and defined by your installation. (For a list of general resource classes supplied by IBM, see Appendix B, “Description of RACF Classes” on page 507.) The member name you specify with the ADDMEM operand can contain one or more generic characters (% or * or **). See Appendix A, “Naming Considerations for Resource Profiles” for information on using generic characters.

• **When SECDATA is the class name**

Security Classification of Users and Data: To define a security category or security level for your installation, specify *class-name* as SECDATA and *profile-name* as one of the following:

- CATEGORY when defining a security category
- SECLEVEL when defining a security level.

If you specify SECDATA CATEGORY, the ADDMEM operand specifies the name of an installation-defined category of users.

For example, to define three categories of users named CODE, TEST, and DOC, enter:

```
RDEFINE SECDATA CATEGORY ADDMEM(CODE TEST DOC)
```

RDEFINE

If you specify SECDATA SECLEVEL, the ADDMEM operand specifies both the name of an installation-defined security level and the number you assign to that level, in the form:

`secl-level-name/secl-level-number`

You must separate the two items by a slash (/). The *secl-level-name* can contain 1-44 characters and must not contain a blank, comma, semicolon, right parenthesis. The *secl-level-number* can be any number from 1 through 254. The higher the number, the higher the security level. For example, to define three security levels, where CONFIDENTIAL is the most restrictive, enter:

```
RDEFINE SECDATA SECLEVEL +  
  ADDMEM(GENERAL/10 EXPERIMENTAL/75 CONFIDENTIAL/150)
```

Because RACF keeps track of security levels by number, replacing an existing security level name does not affect the protection that the security level number provides. If you had defined the security levels shown in the preceding example and then replaced GENERAL/10 with INTERNAL/10, a listing of a user or resource profile that included security level 10 would show the new name. Because the security level number is the same, there is no need to change any resource or user profiles.

When you actually change an existing CATEGORY profile or SECLEVEL profile, however, RACF issues a warning message to remind you that the change is not reflected in existing resource or user profiles. In this case, you can use the SEARCH command to locate the profiles you must modify.

- **When NODES is the class-name**

Translation of User IDs, Group Name, or Security Labels on Inbound Jobs or SYSOUT:

If the class-name is NODES, you can specify how user IDs, group names, and security labels are translated. The translation depends on the second and third qualifiers of the profile name, as follows:

If the Second Qualifier Is:	The ADDMEM Value Specifies:
RUSER	The user ID to be used on this system for the jobs originating from NJE nodes to which the profile applies
USERJ	The user ID to be used on this system for the inbound jobs to which the profile applies
USERS	The user ID to be used on this system for the inbound SYSOUT to which the profile applies
GROUPJ	The group name to be used on this system for the inbound jobs to which the profile applies
GROUPS	The group name to be used on this system for the inbound SYSOUT to which the profile applies
SECLJ	The security label to be used on this system for the inbound jobs to which the profile applies
SECLS	The security label to be used on this system for the inbound SYSOUT to which the profile applies

For information on setting up NODES profiles, see *z/OS Security Server RACF Security Administrator's Guide*.

- **When PROGRAM is the class-name**

Program Control: If you specify *class-name* as PROGRAM, *profile-name* must identify one or more controlled programs (load modules or program

objects), and *member* identifies the library containing the programs, the volume serial of that library, and a processing option. Additionally, APPLDATA may contain information that RACF will process. You specify the member entry in the following format: `library-name/volume-serial/PADCHK` or `NOPADCHK`

library-name

specifies the name of the library in which the controlled programs reside. If *profile-name* is * or **, RACF treats all load modules in the specified library as controlled programs.

If it is necessary to define a program that resides in the system's LPA or dynamic LPA as a controlled program (for example: to give it the MAIN or BASIC attribute), define the program with a *profile-name* that does not end in *, specify 'LPALST' as the library name, and omit the volume serial.

When it is necessary to define a specific profile for a program in the LPA, 'LPALST' should be used as the library name and the volume serial should be omitted.

The following represent valid ADDMEM values for program XYZ which exists in one of the LPA libraries or in the dynamic LPA:

- 'LPALST'
- 'LPALST'//PADCHK
- 'LPALST'//NOPADCHK

volume-serial (optional)

specifies the serial number of the volume on which the library resides. You can use six asterisks within single quotation marks to specify the current SYSRES volume: `library-name/'*****'/PADCHK` or `NOPADCHK`

Notes:

1. '*****' works when the SYSRES resides on more than one volume, but it applies only when the data set lives on the IPL volume
2. If volume-serial is not specified, the specified library can exist on any volume. The alternate formats are:

```
library-name//NOPADCHK
or
library-name
```

PADCHK | NOPADCHK

specifies that RACF is to make (PADCHK) or not to make (NOPADCHK) the checks for program-accessed data sets when a user is executing the controlled programs. If you specify PADCHK, RACF verifies that (1) the conditional access list in the profile for a program-accessed data set allows the access and (2) no task in the user's address space has previously loaded a non-controlled program.

If you specify NOPADCHK, RACF does not perform this extra checking to verify that a non-controlled program cannot access a program-accessed data set. NOPADCHK allows you, for example, to define entire libraries of modules (such as ISPF) as controlled programs without then having to grant each of these modules access to many program-accessed data sets. "Examples" on page 368 show two ways to define controlled programs. Before defining or modifying PROGRAM profiles please read the program control sections of the *z/OS Security Server RACF Security Administrator's Guide*.

RDEFINE

APPLDATA('application-data')

Specifies a text string that is associated with each of the named resources. The text string can contain a maximum of 255 characters and must be enclosed in single quotes. It can also contain double-byte character set (DBCS) data.

Rules:

- For profiles in the PROGRAM class, RACF will examine the APPLDATA (if any) and perform special processing if you have specified 'MAIN' or 'BASIC' (optionally followed by blanks).
 - This processing will occur only for profiles whose names do not end in *, and only when you have enabled Enhanced PGMSECURITY mode.
 - For details of this processing, see *z/OS Security Server RACF Security Administrator's Guide*.
- For the FACILITY class, RACF examines the APPLDATA value for these profiles:
 - BPX.DEFAULT.USER, the APPLDATA specifies a user ID and group name from which RACF can retrieve default OMVS segment information.
 - BPX.NEXT.USER, the APPLDATA specifies information that RACF will use for the automatic assignment of OMVS UIDs and GIDs.
 - IRR.PGMSECURITY, the APPLDATA specifies whether RACF will operate in basic, enhanced, or enhanced-warning PGMSECURITY mode.
 - If the APPLDATA is exactly 'ENHANCED' then RACF will run in enhanced PGMSECURITY mode.
 - If the APPLDATA is exactly 'BASIC' then RACF will run in basic PGMSECURITY mode
 - If the APPLDATA is empty or contains any other value, RACF will run in enhanced PGMSECURITY mode but in warning mode rather than failure mode.
- For the TIMS and GIMS class, specify *application-data* as REVERIFY to force the user to reenter his password whenever the transaction or transactions listed in the *profile-name* or ADDMEM operands are used.
- For the PTKTDATA class, the *application-data* field can be used to control the replay protection function of PassTicket support.

- PassTicket replay protection prevents the use of user IDs to be shared among multiple users. However, in some events it is desirable to bypass this replay protection function.

- Specifying "no replay protection" in the *application-data* field indicates that replay protection is to be bypassed. For example:

```
RDEFINE PTKTDATA profile-name APPLDATA('NO REPLAY PROTECTION')
```

would result in replay protection being bypassed.

Note the following:

- There **must** be a single space between the words "no" and "replay", and "replay" and "protection". Lack of spaces **or** additional spaces or characters will make the command ineffective. For example, entering

```
RDEFINE PTKTDATA profile-name APPLDATA('NOREPLAY PROTECTION')
```

would result in replay protection not being bypassed.

- The text string 'no replay protection' will always be rolled to upper case.
- The text string 'no replay protection' can appear anywhere in the APPLDATA field.

- See *z/OS Security Server RACF Security Administrator's Guide* for more information on the PassTicket function.
- RACF will not validate the APPLDATA during RDEFINE and will not issue any messages during the subsequent processing if it finds an unexpected value.
- This information, if present, can be displayed with the RLIST command. See *z/OS Security Server RACF Security Administrator's Guide* for more information on these APPLDATA values.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([node].userid ...)

specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT([node].userid ...)

specifies that the command is to be directed only to the node specified by *node* where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed only to the local node.

AUDIT(access-attempt[(audit-access-level)])*access-attempt*

specifies which access attempts you want to log on the SMF data set. The following options are available:

ALL

indicates that you want to log both authorized accesses and detected unauthorized access attempts.

FAILURES

indicates that you want to log detected unauthorized access attempts.

NONE

indicates that you do not want any logging to be done.

SUCCESS

indicates that you want to log authorized accesses to the resource.

audit-access-level

specifies which access levels you want to log on the SMF data set. The levels you can specify are:

ALTER

logs ALTER access-level attempts only.

CONTROL

logs access attempts at the CONTROL and ALTER levels.

READ

logs access attempts at any level. This is the default value if no access level is specified.

UPDATE

logs access attempts at the UPDATE, CONTROL, and ALTER levels.

RDEFINE

FAILURES(READ) is the default value if the AUDIT operand is omitted from the command.

You cannot audit access attempts for the EXECUTE level.

DATA('installation-defined-data')

specifies up to 255 characters of installation-defined data to be stored in the profile for the resource and the data must be enclosed in single quotes. It can also contain double-byte character set (DBCS) data.

This information is listed by the RLIST command.

DLFDATA

specifies information used in the control of DLF objects in profiles in the DLFCLASS.

RETAIN(YES | NO)

specifies whether the DLF object can be retained after use

JOBNAMES(jobname-1)

specifies the list of objects which can access the DLF objects protected by this profile.

You can specify any job name valid on your system. You can also specify generic job names with an asterisk (*) as the last character of the job name. For example, JOBNAMES(ABC) allows only job ABC to access the DLF objects protected by the profile. JOBNAMES(ABC*) allows any job whose name begins with ABC (such as ABC, ABC1, or ABCDEF and so forth) to access the DLF objects.

If DLFDATA is not specified, or is specified without the RETAIN suboperand, RETAIN(NO) is defaulted.

EIM

The EIM keyword with the DOMAINDN, OPTIONS, and LOCALREGISTRY subkeywords combined with the PROXY segment define the EIM domain, LDAP host it resides on, and bind information required by the EIM services to establish a connection with an EIM domain. The EIM services will attempt to retrieve this information when it is not explicitly supplied via invocation parameters.

DOMAINDN(eim_domain_dn)

Specifies the distinguished name of the EIM domain. A valid EIM domain distinguished name begins with `ibm-eimDomainName=`. Uppercase and lowercase are accepted and maintained in the case in which they are entered. The EIM domain distinguished name is one component of an EIM domain name. An EIM domain name identifies the LDAP server that stores the EIM domain information. The EIM domain name begins with the `ldap_url` from the LDAPHOST suboperand of the PROXY keyword, followed by `/"` and ends with the `eim_domain_dn` from the DOMAINDN suboperand. The length of a valid EIM domain name is determined by the combination of those factors. RACF allows the input of 1023 characters for the domain distinguished name. RACF does not ensure that an EIM domain name created from the LDAP URL and EIM domain distinguished name forms a valid EIM domain name. See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP distinguished names.

OPTIONS

specifies options that control the EIM configuration.

ENABLE|DISABLE**ENABLE**

Specifies that new connections may be established with the specified EIM domain. This is the default.

DISABLE

Specifies that new connections may not be established with the specified EIM domain.

LOCALREGISTRY(*registry_name*)

Specifies the name of the local RACF registry in EIM domains. This operand is only valid for the IRR.PROXY.DEFAULTS FACILITY class profile. The value is ignored when used on other profiles. Changing this value does not affect the in-storage copy of the registry name until SETROPTS EIMREGISTRY is issued or the system is IPLed. The local *registry_name* may be 1 to 255 characters long and is not case sensitive. The local registry name is stored in the RACF database in uppercase.

FCLASS(*profile-name-2-class*)

specifies the name of the class to which *profile-name-2* belongs. The valid class names are DATASET and those classes defined in the class descriptor table. For a list of general resource classes defined in the class descriptor table supplied by IBM, go to page Appendix B, “Description of RACF Classes” on page 507.

If you omit this operand, RACF assumes that *profile-name-2* belongs to the same class as *profile-name-1*. This operand is valid only when you also specify the FROM operand; otherwise, RACF ignores it.

FGENERIC

specifies that RACF is to treat *profile-name-2* as a generic name, even if it is fully-qualified (meaning that it does not contain any generic characters). This operand is needed only if *profile-name-2* is a DATASET profile.

FROM(*profile-name-2*)

specifies the name of an existing discrete or generic profile that RACF is to use as a model for the new profile. The model profile name you specify on the FROM operand overrides any model name specified in your user or group profile. If you specify FROM and omit FCLASS, RACF assumes that *profile-name-2* is the name of a profile in the same class as *profile-name-1*.

Mixed case profile names are accepted and preserved when FCLASS refers to a class defined in the class descriptor table with CASE=ASIS.

To specify FROM, you must have sufficient authority to both *profile-name-1* and *profile-name-2*, as described under “Authorization Required.”

Possible Changes to Copied Profiles When Modeling Occurs

When a profile is copied during profile modeling, the new profile might differ from the model in the following ways:

- RACF places the user’s ID on the access list with ALTER access authority or, if the user’s ID is already on the access list, RACF changes the user’s access authority to ALTER. If NOADDCREATOR is in effect, however, RACF copies the access list authorities exactly as they appear in the model’s access list.
- If the model profile contains members (specified with the ADDMEM operand), the members are not copied into the new profile.

RDEFINE

- If the SETROPTS MLS option is in effect, the security label (if specified in the model profile) is not copied. Instead, the user's current security label is used.

EXCEPTION: When SETROPTS MLS and MLSTABLE are both in effect and the user has the SPECIAL attribute, the security label specified in the model profile is copied to the new profile.

- For TAPEVOL profiles, TVTOC information is not copied to the new profile.
- Information in the non-RACF segments (for example, the SESSION or DLFDATA segment) is not copied.

For information about automatic profile modeling, refer to *z/OS Security Server RACF Security Administrator's Guide*.

FVOLUME(*volume-serial*)

specifies the volume RACF is to use to locate the model profile (*profile-name-2*).

If you specify FVOLUME and RACF does not find *profile-name-2* associated with that volume, the command fails. If you omit this operand and *profile-name-2* appears more than once in the RACF data set, the command fails.

FVOLUME is valid only when FCLASS either specifies or defaults to DATASET and when *profile-name-2* specifies a discrete profile. Otherwise, RACF ignores FVOLUME.

KERB

specifies Security Server Network Authentication Service information for a REALM class profile.

DEFTKTLFE(*def-ticket-life*)

specifies the default ticket lifetime for the local Network Authentication Service in seconds. DEFTKTLFE is a numeric value between 1 and 2 147 483 647. Note that 0 is not a valid value.

This keyword is only applicable when defining the KERBDFLT REALM profile for the local realm.

If DEFTKTLFE is specified, MAXTKTLFE and MINTKTLFE must also be specified.

ENCRYPT [[([**DES** | **NODES**] [**DES3** | **NODES3**] [**DESD** | **NODESD**])]

The ENCRYPT values are used to specify which keys are allowed for use based on the encryption algorithm used to generate them. The default values for ENCRYPT are DES, DES3, and DESD. You can use the following values to specify which keys are allowed for use by a principal.

DES	DES encrypted keys are allowed for use.
NODES	No DES encrypted keys are allowed for use.
DES3	DES3 encrypted keys are allowed for use.
NODES3	No DES3 encrypted keys are allowed for use.
DESD	DESD encrypted keys are allowed for use.
NODESD	No DESD encrypted keys are allowed for use.

The values in effect are dependent on the current SETROPTS KERBLVL setting.

When SETROPTS KERBLVL(0) is in effect the ENCRYPT settings will be ignored. Regardless of the settings DES keys will be generated and processed.

When SETROPTS KERBLVL(1) is in effect, or when SETROPTS KERBLVL gets changed from 0 to 1, the ENCRYPT settings will go into effect. Therefore, on password change, all three keys are generated and stored in the user's profile. The ENCRYPT setting will be used to determine which keys can be processed.

If you do not want to accept the defaults, you must specify the values you desire. For example, if you want to use only DES3 encryption, you must specify ENCRYPT(NODES DES3 NODESD).

If you specify ENCRYPT(NODES, NODES3, NODESD) at KERBLVL(1), no keys can be used, but all three will be generated and stored. At KERBLVL(0), the DES key will still be generated and it cannot be disallowed.

KERBNAME(*kerberos-realm-name*)

specifies the unqualified name of the local Kerberos realm for Network Authentication Service. The maximum length of this field is 117 characters. The fully qualified form of the local Kerberos realm name

```
/.../kerberos_realm_name/krbtgt/local_realm_name
```

must not be specified.

The name assigned to the local realm limits the length of local principal names, since fully qualified local principal names

```
/.../kerberos_realm_name/principal_name
```

cannot exceed 240 characters.

The local realm name that you define to RACF can consist of any character, except the / (X'61') character. It is highly recommended that you avoid using **any** of the EBCDIC variant characters be avoided to prevent problems with different code pages. You can enter the name with or without single quotes, depending on the following:

- If parentheses, commas, blanks, or semicolons are entered as part of the name, the character string must be enclosed in single quotes.
- If a single quote is intended to be part of the name and the entire character string is enclosed in single quotes, you must use two single quotes together to represent each single quote within the string.
- If the first character of the name is a single quote, you must enter the string within single quotes, with two single quotes entered for the single quote.

Regardless of the case in which it is entered, RACF rolls the name of the local Network Authentication Service realm to upper case. However, RACF does not ensure that a valid *kerberos-realm-name* has been specified.

Note: Because of the relationship between the realm name and the local principal name where the length of a fully qualified name cannot exceed 240 characters, caution and planning must go into renaming the local realm since the combined length is only checked by RACF when a local *kerberos-realm-name* is added or altered. Renaming the realm should be avoided as a result. In addition, if the name of the realm does change, the user's keys will become unusable.

RDEFINE

This keyword is only applicable when defining the KERBDFLT REALM profile for the local Network Authentication Service realm.

MAXTKLFE(*max-ticket-life*)

specifies the *max-ticket-life* for the local Network Authentication Service in seconds. MAXTKLFE is a numeric value between 1 and 2 147 483 647. Note that 0 is not a valid value.

This keyword is only applicable when defining the KERBDFLT REALM profile for the local Network Authentication Service realm.

If MAXTKLFE is specified, DEFTKLFE and MINTKLFE must also be specified.

MINTKLFE(*min-ticket-life*)

specifies the *min-ticket-life* for the Network Authentication Service in seconds. MINTKLFE is a numeric value between 1 and 2 147 483 647. Note that 0 is not a valid value.

This keyword is only applicable when defining the KERBDFLT REALM profile for the local Kerberos realm.

If MINTKLFE is specified, DEFTKLFE and MAXTKLFE must also be specified.

PASSWORD(*kerberos-password*)

Specifies the value of the *kerberos-password*. The maximum length of this value is 8 characters. The PASSWORD keyword is applicable to all REALM class profile definitions. A password is required in order for the local realm to grant ticket-granting tickets and a password must be associated with the definition of a foreign realm inter-realm trust relationship, or the definition is incomplete.

The password that you define to RACF can consist of any character. It is highly recommended that use of **any** of the EBCDIC variant characters be avoided, to prevent problems with different code pages. You can enter a password with or without single quotes, depending on the following:

- If parentheses, commas, blanks, or semicolons are entered as part of the password, the character string must be enclosed in single quotes.
- If a single quote is intended to be part of the password and the entire character string is enclosed in single quotes, you must use two single quotes together for each single quote within the string.
- If the first character of the password is a single quote, you must enter the string within single quotes, with two single quotes entered for the character.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered.

Note: This keyword is intended for administrators to be able to associate a *kerberos-password* with the definition of a realm. It is not the same as a RACF user password and is not constrained by the SETROPTS password rules or password interval values that might be established for RACF user passwords.

LEVEL(*nn*)

specifies a level indicator, where *nn* is an integer between 0 and 99. The default is 0.

Your installation assigns the meaning of the value. It is included on all records that log resource accesses and is listed by the RLIST command.

NOTIFY[(userid)]

specifies the user ID of a user to be notified whenever RACF uses this profile to deny access to a resource. If you specify NOTIFY without specifying a user ID, RACF takes your user ID as the default; you are notified whenever the profile denies access to a resource.

A user who is to receive NOTIFY messages should log on frequently to take action in response to the unauthorized access attempt described in each message. RACF sends NOTIFY messages to the SYS1.BROADCAST data set. When the resource profile also includes WARNING, RACF might have granted access to the resource to the user identified in the message.

When RACF denies access to a resource, it does **not** notify a user:

- When the resource is in the PROGRAM class
- When the resource is in a class for which an application has built in-storage profiles using RACROUTE REQUEST=LIST

Some applications, such as IMS and CICS, load all the profiles for a given class into storage. After these profiles are in storage, the applications can do a “fast” authorization check using RACROUTE REQUEST=FASTAUTH. One difference is that, in some cases, fast authorization checking does not issue warning messages, notification messages, or support auditing. In cases where it does not, return and reason codes are returned to the application to allow support of these functions. Return and reason codes are returned to the application to allow support of these functions. The application can examine the return and reason codes and use RACROUTE REQUEST=AUTH to create the messages and audit records. If the application uses RACROUTE REQUEST=AUTH to support auditing, the specified user is notified. Otherwise, notification, warning, and such does not occur.

For details on using RACF with IMS, see *z/OS Security Server RACF Security Administrator's Guide*. For details on using RACF with CICS, refer *CICS RACF Security Guide*.

- When the profile is used to disallow the creation or deletion of a data set
NOTIFY is used only for resource access checking, not for resource creation or deletion.

OWNER(userid or group-name)

specifies a RACF-defined user or group to be assigned as the owner of the resource you are defining. If you omit this operand, you are defined as the owner. The user specified as the owner does not automatically have access to the resource. Use the PERMIT command to add the owner to the access list as desired.

PROXY

specifies information which the z/OS LDAP Server will use when acting as a proxy on behalf of a requester. The R_proxyserv (IRRSPY00) SAF callable service will attempt to retrieve this information when it is not explicitly supplied via invocation parameters. Applications or other services which use the R_proxyserv callable service, such as IBM Policy Director Authorization Services for z/OS and OS/390, may instruct their invokers to define PROXY segment information.

LDAPHOST(ldap_url)

specifies the URL of the LDAP server which the z/OS LDAP Server will

RDEFINE

contact when acting as a proxy on behalf of a requester. An LDAP URL has a format such as `ldap://123.45.6:389` or `ldaps://123.45.6:636`, where `ldaps` indicates that an SSL connection is desired for a higher level of security. LDAP will also allow you to specify the host name portion of the URL using either the text form (`BIGHOST.POK.IBM.COM`) or the dotted decimal address (`123.45.6`). The port number is appended to the host name, separated by a colon ':' (`X'7A'`). See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP URLs and how to enable LDAP servers for SSL connections.

The LDAP URL that you define to RACF can consist of 10—1023 characters. A valid URL must start with either `ldap://` or `ldaps://`. RACF will allow any characters to be entered for the remaining portion of the URL, but you should ensure that the URL conforms to TCP/IP conventions. For example, parentheses, commas, blanks, semicolons, and single quotes are not typically allowed in a host name. The LDAP URL can be entered with or without single quotes, however, in both cases, it will be folded to uppercase.

RACF does not ensure that a valid LDAP URL has been specified.

BINDDN(*bind_distinguished_name*)

specifies the distinguished name (DN) which the z/OS LDAP Server will use when acting as a proxy on behalf of a requester. This DN will be used in conjunction with the BIND password, if the z/OS LDAP Server needs to supply an administrator or user identity to BIND with another LDAP Server. A DN is made up of attribute value pairs, separated by commas. For example:

```
cn=Ben Gray,ou=editing,o=New York Times,c=US
cn=Lucille White,ou=editing,o=New York Times,c=US
cn=Tom Brown,ou=reporting,o=New York Times,c=US
```

See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP DNs.

When you define a BIND DN to RACF, it can contain 1—1023 characters. The BIND DN can consist of any characters and can be entered with or without single quotes. The following rules apply:

- If parentheses, commas, blanks, or semicolons are to be entered as part of the BIND DN, the character string must be enclosed in single quotes.
- If a single quote is intended to be part of the BIND DN, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered. See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP distinguished names.

If you issue the RDEFINE command as a RACF operator command and you specify the BIND DN in lowercase, you must include the BIND DN within single quotations.

RACF does not ensure that a valid BIND DN has been specified.

BINDPW

specifies the password which the z/OS LDAP Server will use when acting as a proxy on behalf of a requester.

When you define a BIND password to RACF, it can contain 1-128 characters. The BIND password can consist of any characters (see exception below) and can be entered with or without single quotes. The following rules apply:

- The BIND password can not start with a left curly brace '{' (X'8B').
- If parentheses, commas, blanks, or semicolons are to be entered as part of the BIND password, the character string must be enclosed in single quotes.
- If a single quote is intended to be part of the BIND password, you must use two single quotes together for each single quote within the string, and the entire string must be enclosed within single quotes.

Both uppercase and lowercase are accepted and maintained in the case in which they are entered. See *z/OS Security Server LDAP Server Administration and Use* for more information about LDAP passwords.

If you issue the RDEFINE command as a RACF operator command and you specify the BIND password in lowercase, you must include the BIND password within single quotations.

RACF does not ensure that a valid BIND password has been specified.

Attention:

- When the command is issued from ISPF, the TSO command buffer (including possible BINDPW password data) is written to the ISPLOG data set. As a result, you should not issue this command from ISPF or you must control the ISPLOG data set carefully.
- When the command is issued as a RACF operator command, the command and the possible BINDPW password data is written to the system log. Therefore, use of RALTER as a RACF operator command should either be controlled or you should issue the command as a TSO command.

SECLABEL(*seclabel-name*)

specifies the user's resource's default security label, where *seclabel-name* is an installation-defined security label name that represents an association between a particular security level and a set of zero or more categories.

A security label corresponds to a particular security level (such as CONFIDENTIAL) with a set of zero or more security categories (such as PAYROLL or PERSONNEL).

RACF stores the name of the security label you specify in the resource profile if you are authorized to use that SECLABEL.

If you are not authorized to the SECLABEL or if the name you had specified is not defined as a SECLABEL profile in the SECLABEL class, the resource profile is not created.

SECLEVEL(*seclabel-name*)

specifies the name of an installation-defined security level. The name corresponds to the number that is the minimum security level that a user must have to access the resource. The *seclabel-name* must be a member of the SECLEVEL profile in the SECDATA class.

When you specify SECLEVEL and the SECDATA class is active, RACF adds security level checking to its other authorization checking. If global access checking grants access, RACF compares the security level allowed in the user

RDEFINE

profile with the security level required in the resource profile. If the security level in the user profile is less than the security level in the resource profile, RACF denies the access. If the security level in the user profile is equal to or greater than the security level in the resource profile, RACF continues with other authorization checking. The SECLEVEL operand is required for the SECLABEL class.

Note: RACF does not perform security level checking for a started task that has the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF started procedures table or STARTED class. Also, RACF does not enforce security level information specified on profiles in the PROGRAM class.

If the SECDATA class is not active, RACF stores the name you specify in the resource profile. When the SECDATA class is activated and the name you specified is defined as a SECLEVEL profile, RACF can perform security level access checking for the resource profile. If the name you specify is not defined as a SECLEVEL profile, you are prompted to provide a valid SECLEVEL name.

SESSION

is only valid for the APPCLU resource class. It specifies that when changing an APPCLU class profile, the following suboperands add, change, or delete SESSION segment field values. The SESSION segment is used to control the establishment of sessions between logical units under LU6.2.

CONVSEC

specifies the level or levels of security checking performed when conversations are established with the LU protected by this profile.

Note: In general, you should select one of ALREADYV, AVPV, CONV, NONE or PERSISTV for each APPCLU profile.

ALREADYV

APPC/MVS RACF does *not* verify the user ID and password for any inbound allocate requests. If you specify ALREADYV, you assume that user IDs and passwords have already been verified by the partner LU. You must specify this only if the partner LU is trustworthy.

AVPV

The user ID/password is already verified and persistent verification is requested.

CONV

APPC/MVS issues a RACROUTE REQUEST=VERIFY to verify the user ID and password for all inbound allocate requests.

NONE

All inbound allocate requests pass without RACF checking for a valid user ID. No RACROUTE REQUEST=VERIFY is issued.

PERSISTV

Specifies persistent verification.

INTERVAL(*n*)

sets the maximum number of days the session key is valid. The variable *n* is in the range of 1 to 32767. If the key interval is longer than the installation maximum (set with SETROPTS SESSIONINTERVAL), then the profile is created.

If the key interval is not specified and there is a SETROPTS SESSIONINTERVAL value, the profile is created with that value. If there is no SETROPTS SESSIONINTERVAL value, there is no limit to the number of days the session key is valid.

LOCK

mark the profile as locked. This prevents all session establishment from succeeding.

SESSKEY(*session-key*)

change the key for this profile. The variable *session-key* can be expressed in two ways:

- X'y'—where y is a 1-16-digit hexadecimal number
- z or 'z'—where z is a 1-8 character string

If the entire 16 digits or 8 characters are not used, the field is padded to the right with binary zeros.

Note: Session keys are 64-bit Data Encryption Standard (DES) keys. With DES, 8 of the 64 bits are reserved for use as parity bits, so those 8 bits are not part of the 56-bit key. In hexadecimal notation, the DES parity bits are:X'0101 0101 0101 0101'. Any two 64-bit keys are equivalent DES keys if their only difference is in one or more of these parity bits. For instance, the following SESSKEY values, although appearing to be quite different, are equivalent because they differ only in the last bit of each byte:

- BDF0KM4Q, which is X'C2C4 C6F0 D2D4 F4D8'
- CEG1LN5R, which is X'C3C5 C7F1 D3D5 F5D9'

SINGLEDSN

specifies that the tape volume can contain only one data set. SINGLEDSN is valid only for a TAPEVOL profile. If the volume already contains more than one data set, RACF issues a message and ignores the operand.

SSIGNON(**KEYMASKED**(*key-value*))(**KEYENCRYPTED**(*key-value*))

defines the application key or a secured signon key and indicates the method you want to use to protect the key value within the RACF database on the host. When defining the profile, you can either mask or encrypt the key. The *key-value* represents a 64-bit (8-byte) key that must be represented as 16 hexadecimal characters. The valid characters are 0 through 9 and A through F.

Notes:

1. As with RACF passwords, the database unload facility does not unload application keys or the secured signon keys.
2. The RLIST command does not list the value of the application keys or a secured signon keys. Therefore, when you define the keys, you should note the value and keep it in a secure place.

KEYMASKED(*key-value*)

indicates that you want to mask the key value using the masking algorithm.

Rules:

- You can specify this operand only once for each application key.
- If you mask a key, you *cannot* encrypt it. These are mutually exclusive.

You can use the RLIST command to ensure that the key is protected.

RDEFINE

KEYENCRYPTED(*key-value*)

indicates that you want to encrypt the key value.

Rules:

- You can specify this operand only once for each application key.
- If you encrypt a key, you *cannot* mask it. These are mutually exclusive.
- A cryptographic product must be installed and active on the system.

You can use the RLIST command to verify that the key is protected.

STDATA

used to control security for started tasks. STDATA should only be specified for profiles in the STARTED class.

USER(*userid* | **=MEMBER**)

USER(*userid*)

specifies the user ID to be associated with this entry.

RACF issues a warning message if the specified *userid* does not exist, or if the USER operand is not specified, but data is placed into the STDATA segment. If the error is not corrected, RACF uses the started procedures table to process START requests that would have used this STARTED profile.

USER(**=MEMBER**)

specifies that the procedure name should be used as the user ID. If **=MEMBER** is specified for USER, a *group-name* value should be specified for the GROUP operand. If **=MEMBER** is specified for both USER and GROUP, a warning message is issued and problems might result when the profile is used. For information, see *z/OS Security Server RACF System Programmer's Guide*.

GROUP(*group-name* | **=MEMBER**)

GROUP(*group-name*)

specifies the group name to be associated with this entry.

RACF issues a warning message if the specified *group-name* does not exist. If *userid* and *group-name* are specified, RACF verifies that the user is connected to the group. If GROUP is specified incorrectly, the started task runs as an undefined user.

GROUP(**=MEMBER**)

specifies that the procedure name should be used as the group name. If **=MEMBER** is specified for GROUP, a *userid* value must be specified for the USER operand or RACF uses the started procedures table to assign an identity for this started task. If **=MEMBER** is specified for both USER and GROUP, a warning message is issued and problems might result when the profile is used. For information, see *z/OS Security Server RACF System Programmer's Guide*.

If GROUP is not specified the started task runs with the default group of the specified user ID.

PRIVILEGED(**NO** | **YES**)

specifies whether the started task should run with the RACF PRIVILEGED attribute. The PRIVILEGED attribute allows the started task to pass most authorization checking. No installation exits are called, no SMF records are

generated, and no statistics are updated. (Note that bypassing authorization checking includes bypassing the checks for security classification of users and data.) For more information, see *z/OS Security Server RACF System Programmer's Guide*.

If PRIVILEGED(NO) is specified, the started tasks runs without the RACF PRIVILEGED attribute.

If PRIVILEGED is not specified PRIVILEGED(NO) is defaulted.

TRACE(NO | YES)

specifies whether a message should be issued to the operator when this entry is used to assign an ID to the started task.

If TRACE(YES) is specified, RACF issues an informational message to the operator to record the use of this entry when it is used to assign an ID to a started task. This record can be useful in finding started tasks that do not have a specific entry defined and in diagnosing problems with the user IDs assigned for started tasks.

If TRACE(NO) is specified, RACF does not issue an informational message when this entry is used.

If TRACE is not specified, TRACE(NO) is defaulted.

TRUSTED(NO | YES)

specifies whether the started task should run with the RACF TRUSTED attribute. The TRUSTED attribute is similar to the PRIVILEGED attribute except that auditing can be requested using the SETROPTS LOGOPTIONS command. For more information about the TRUSTED attribute, see *z/OS Security Server RACF System Programmer's Guide*.

If TRUSTED(NO) is specified, the started tasks runs without the RACF TRUSTED attribute.

If TRUSTED is not specified, TRUSTED(NO) is defaulted.

SVFMR

defines profiles associated with a particular SystemView for MVS application.

SCRIPTNAME(*script-name*)

specifies the name of the list of default logon scripts associated with this application. This operand is optional. If this operand is omitted, no scripts are associated with the application.

The *script-name* is a 1-8 character alphanumeric name of a member of an MVS partitioned data set (PDS). RACF accepts both upper- and lowercase characters for *script-name*, but lowercase characters are translated to uppercase.

The PDS member specified by the *script-name* contains a list of other PDS members that contain the scripts associated with this application's profile. The PDS and members, including the member that contains the list of other members, are created by the SystemView for MVS administrator.

PARMNAME(*parm-name*)

specifies the name of the parameter list associated with this application. If this operand is omitted, no parameters are associated with the application.

The *parm-name* is a 1-8 character alphanumeric name of a member of an MVS partitioned data set (PDS). RACF accepts both upper and lowercase characters for *parm-name*, but lowercase characters are translated to uppercase.

RDEFINE

The PDS member specified by *parm-name* contains a list of other PDS members that contain the parameters associated with this application's profile. The PDS and members, including the list of other members, are created by System View for the MVS administrator.

TIMEZONE({E | W} *hh* [*mm*])

specifies the time zone in which a terminal resides. TIMEZONE is valid only for resources in the TERMINAL class; RACF ignores it for all other resources.

Specify TIMEZONE only when the terminal is not in the same time zone as the processor on which RACF is running and you are also specifying WHEN to limit access to the terminal to specific time periods. In this situation, TIMEZONE provides the information RACF needs to calculate the time values correctly. If you identify more than one terminal in the *profile-name-1* operand, all the terminals must be in the same time zone.

On TIMEZONE, you specify whether the terminal is east (E) or west (W) of the system and by how many hours (*hh*) and, optionally, minutes (*mm*) that the terminal time zone is different from the processor time zone. Valid hour values are 0 through 11, and valid minute values are 00 through 59.

For example, if the processor is in New York and the terminal is in Los Angeles, specify TIMEZONE(W 3). If the processor is in Houston and the terminal is in New York, specify TIMEZONE(E 1).

If you change the local time on the processor (to accommodate daylight savings time, for instance), RACF adjusts its time calculations accordingly. If, however, the processor time zone and the terminal time zone do not change in the same way, you must adjust the terminal time zones yourself, as described earlier for the WHEN(TIME) operand.

TME

specifies that information for the Tivoli Security Management Application be added.

Note: The TME segment fields are intended to be updated only by the Tivoli Security Management Application, which manages updates, permissions, and cross references. A security administrator should only directly update Tivoli Security Management fields on an exception basis.

All TME suboperands, with the exception of those for ROLES, can be specified when changing a resource profile in the ROLE class. Conversely, only the ROLES suboperands can be specified when changing a resource profile in any other class.

CHILDREN(*profile-name* ...)

specifies the complete list of roles that inherit attributes from this role. A role is a discrete general resource profile defined in the ROLE class.

GROUPS(*group-name* ...)

specifies the complete list of groups that should be permitted to resources defined in this role profile.

PARENT(*profile-name*)

specifies the name of a role from which this role inherits attributes. A role is a discrete general resource profile defined in the ROLE class.

RESOURCE(*resource-access-specification* ...)

specifies the complete list of resources and associated access levels for groups defined in this role profile.

One or more *resource-access-specification* values can be specified, each separated by blanks. Each value should contain no imbedded blanks and should have the following format:

origin-role: class-name: profile-name: access-authority [: conditional-class: conditional-profile]

where *origin-role* is the name of the role profile from which the resource access is inherited. *Class-name* is an existing resource class name and *profile-name* is a resource profile defined in that class. The *access-authority* is the authority (NONE, EXECUTE, READ, UPDATE, CONTROL, or ALTER) with which groups in the role definition should be permitted to the resource.

The *conditional-class* is a class name (APPCPORT, CONSOLE, JESINPUT, PROGRAM, TERMINAL, or SYSID) for conditional access permission, and is followed by the *conditional-profile* value, a resource profile defined in the conditional class.

ROLES(*role-access-specification ...*)

specifies a list of roles and associated access levels related to this profile.

One or more *role-access-specification* values can be specified, each separated by blanks. Each value should contain no imbedded blanks and should have the following format:

role-name: access-authority [: conditional-class : conditional-profile]

where *role-name* is a discrete general resource profile defined in the ROLE class. The *access-authority* is the authority (NONE, EXECUTE, READ, UPDATE, CONTROL, or ALTER) with which groups in the role definition should be permitted to the resource.

The *conditional-class* is a class name (APPCPORT, CONSOLE, JESINPUT, PROGRAM, TERMINAL, or SYSID) for conditional access permission, and is followed by the *conditional-profile* value, a resource profile defined in the conditional class.

TVTOC

specifies, for a TAPEVOL profile, that RACF is to create a TVTOC in the TAPEVOL profile when a user creates the first output data set on the volume. The RDEFINE command creates a nonautomatic TAPEVOL profile; RACF creates and maintains the TVTOC for data sets residing on tape.

Specifying TVTOC also affects the access list for the TAPEVOL profile:

1. When RACF processes the RDEFINE command with the TVTOC operand, it places the user ID of the command issuer (perhaps the tape librarian) in the access list with ALTER authority.
2. When the first output data set is created on the volume, RACF adds the user ID associated with the job or task to the access list with ALTER authority.

See *z/OS Security Server RACF Security Administrator's Guide* for further information.

The TVTOC operand is valid only for a discrete profile in the TAPEVOL class.

UACC(*access-authority*)

specifies the universal access authority to be associated with this resource. The universal access authorities are ALTER, CONTROL, UPDATE, READ, EXECUTE (for controlled programs only), and NONE. If UACC is not specified,

RDEFINE

RACF uses the value in the ACEE or the class descriptor table. If UACC is specified without *access-authority*, RACF uses the value in the current connect group. For tape volumes and DASD volumes, RACF treats CONTROL authority as UPDATE authority. For all other resources listed in the class descriptor table and for applications, RACF treats CONTROL and UPDATE authority as READ authority.

If the user ID accessing the general resource has the RESTRICTED attribute, RACF treats the access authority as NONE.

WARNING

specifies that even if access authority is insufficient, RACF is to issue a warning message and allow access to the resource. RACF also records the access attempt in the SMF record if logging is specified in the profile. RACF does *not* issue a warning message for a resource:

- When the resource is in the PROGRAM class
- When the resource is in the NODES class.
- When the resource is in a class for which an application has built in-storage profiles using RACROUTE REQUEST=LIST

Some applications, such as IMS and CICS, load all the profiles for a given class into storage. After these profiles are in storage, the applications can do a “fast” authorization check using RACROUTE REQUEST=FASTAUTH. One difference is that, in some cases, fast authorization checking does not issue warning messages, notification messages, or support auditing. In cases where it does not, return and reason codes are returned to the application to allow support of these functions. Return and reason codes are returned to the application to allow support of these functions. The application can examine the return and reason codes and use RACROUTE REQUEST=AUTH to create the messages and audit records. If the application uses RACROUTE REQUEST=AUTH to support auditing, the specified user is notified. Otherwise, notification, warning, and so on do not occur.

For details on using RACF with IMS, see *z/OS Security Server RACF Security Administrator's Guide*. For details on using RACF with CICS, refer *CICS RACF Security Guide*.

WHEN([DAYS(*day-info*)] [TIME(*time-info*)])

specifies, for a resource in the TERMINAL class, the days of the week or the hours in the day when a user can access the system from the terminal. The day-of-week and time restrictions apply only when a user logs on to the system; that is, RACF does not force the user off the system if the end-time occurs while the user is logged on.

If you omit the WHEN operand, a user can access the system from the terminal at any time. If you specify the WHEN operand, you can restrict the use of the terminal to certain days of the week or to a certain time period on each day. Or, you can restrict access to both certain days of the week and to a certain time period within each day.

DAYS(*day-info*)

To allow use of the terminal only on certain days, specify **DAYS(*day-info*)**, where *day-info* can be any one of the following:

ANYDAY

RACF allows use of the terminal on any day. If you omit DAYS, ANYDAY is the default.

WEEKDAYS

RACF allows use of the terminal only on weekdays (Monday through Friday).

day...

RACF allows use of the terminal only on the days specified, where day can be MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, or SUNDAY. You can specify the days in any order.

TIME(*time-info*)

To allow use of the terminal only during a certain time period of each day, specify **TIME**(*time-info*), where *time-info* can be any one of the following:

ANYTIME

RACF allows use of the terminal at any time. If you omit TIME, ANYTIME is the default.

start-time: end-time

RACF allows use of the terminal only during the specified time period. The format of both start-time and end-time is *hhmm*, where *hh* is the hour in 24-hour notation (00 through 24) and *mm* is the minutes (00 through 59) within the range 0001 - 2400. Note that 2400 indicates 12:00 a.m. (midnight).

If *start-time* is greater than *end-time*, the interval spans midnight and extends into the following day.

Specifying *start-time* and *end-time* is straightforward when the processor on which RACF is running and the terminal are in the same time zone; you specify the time values in local time.

If, however, the terminal is in a different time zone from the processor and you want to restrict access to certain time periods, you have two choices. You can specify the TIMEZONE operand to allow RACF to calculate the time and day values correctly. Otherwise, you can adjust the time values yourself, by translating the *start-time* and *end-time* for the terminal to the equivalent local time for the processor.

For example, assume that the processor is in New York and the terminal is in Los Angeles, and you want to allow access to the terminal from 8:00 A.M. to 5:00 P.M. in Los Angeles. In this situation, you would specify TIME(1100:2000). If the processor is in Houston and the terminal is in New York, you would specify TIME(0900:1800).

If you omit DAYS and specify TIME, the time restriction applies to all seven days of the week. If you specify both DAYS and TIME, RACF allows use of the terminal only during the specified time period and only on the specified days.

RDEFINE

Examples

Table 55. RDEFINE Examples

Example 1	<i>Operation</i>	User TBK20 wants to define resource GIMS600 in class GIMS which is a resource group class. He also wants to define TIMS200, TIMS111, TIMS300, and TIMS333 as members of the resource group (GIMS600).
	<i>Known</i>	User TBK20 has the CLAUTH attribute for the GIMS and TIMS classes. GIMS is a resource group class, and TIMS is its associated resource member class. TIMS200 and TIMS111 are members of another resource group. The user has ALTER authority to the other resource group. User TBK20 wants to issue the command as a RACF TSO command.
	<i>Command</i>	RDEFINE GIMS GIMS600 ADDMEM(TIM S200 TIM S111 TIM S300 TIM S333)
Example 2	<i>Defaults</i>	OWNER (TBK20) LEVEL(0) AUDIT(FAILURES(READ)) UACC(NONE)
	<i>Operation</i>	User ADM1 wants to define a generic profile for all resources starting with a "T" belonging to the TIMS class, and to require that users must reenter their passwords whenever they enter any IMS transaction starting with a T.
	<i>Known</i>	User ADM1 has the SPECIAL attribute. User ADM1 wants to issue the command as a RACF TSO command.
Example 3	<i>Command</i>	RDEFINE TIMS T* APPL('REVERIFY')
	<i>Defaults</i>	UACC(NONE) OWNER(ADM1) LEVEL(0) AUDIT(FAILURES(READ))
	<i>Operation</i>	User ADM1 wants to define AMASPZAP as a controlled program with program-accessed data set checking.
Example 4	<i>Known</i>	User ADM1 has the SPECIAL attribute. AMASPZAP resides in SYS1.LINKLIB on the SYSRES volume. RACF program control is active. User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	RDEFINE PROGRAM AMASPZAP ADDMEM('SYS1.LINKLIB'/SYSRES/PADCHK)
	<i>Defaults</i>	UACC(NONE) OWNER(ADM1) LEVEL(0) AUDIT(FAILURES(READ))
Example 5	<i>Operation</i>	User ADM1 wants to define all load modules that start with IKF as controlled programs that do not require program-accessed data set checking.
	<i>Known</i>	User ADM1 has the SPECIAL attribute. All load modules whose names begin with IKF reside in SYS1.COBLIB on the SYSRES volume. User ADM1 wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.
	<i>Command</i>	@RDEFINE PROGRAM IKF* ADDMEM('SYS1.COBLIB'/SYSRES/NOPADCHK)
Example 6	<i>Defaults</i>	UACC(NONE) OWNER(ADM1) LEVEL(0) AUDIT(FAILURES(READ))
	<i>Operation</i>	User JPQ12 wants to define a tape volume labeled DP0123 and allow it to hold a TVTOC. The tape volume is assigned a UACC of NONE.
	<i>Known</i>	User JPQ12 has the SPECIAL attribute. User JPQ12 wants to issue the command as a RACF TSO command.
Example 7	<i>Command</i>	RDEFINE TAPEVOL DP0123 TVTOC UACC(NONE)
	<i>Defaults</i>	OWNER (JPQ12) LEVEL(0) AUDIT(FAILURES(READ))
	<i>Operation</i>	User ADM1 wants to prepare the TCICSTRN class to be used for RACGLIST processing.
Example 8	<i>Known</i>	User ADM1 has the SPECIAL attribute User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	RDEFINE RACGLIST TCICSTRN UACC(NONE)
	<i>Defaults</i>	OWNER(ADM1) LEVEL(0) AUDIT(FAILURES(READ))

Table 55. RDEFINE Examples (continued)

Example 7

Operation The security administrator wants to define a profile for TSO in the PTKTDATA class. The security administrator wants to direct the command to run under the authority of user OJC11 at node NYTSO.

Known SIVLE1 is the user ID of the security administrator.

OJC11 has the SPECIAL attribute on node NYTSO.

The profile name is TSOR001.

The *key-value* is e001193519561977 and is to be masked. The security administrator wants to issue the command as a RACF TSO command.

The security administrator and OJC11 at NYTSO have an already established user ID association.

Command RDEFINE PTKTDATA TSOR001 SSIGNON(KEYMASKED(e001193519561977))
AT(NYTSO.OJC11)

Defaults UACC(NONE)

Example 8

Operation The security administrator wants to create an entry in the dynamic started procedures table for the OMVS started procedure by defining a generic profile in the STARTED class.

Known The administrator wants to use the procedure name as the user ID. The group name is STCGRP.

SETROPTS GENERIC(STARTED) has been issued to allow generic profiles to be created in this class. The security administrator wants to issue the command as a RACF TSO command.

Command RDEFINE STARTED OMVS.* STDATA(USER(=MEMBER) GROUP(STCGRP))

Defaults PRIVILEGED(NO) TRACE(NO) TRUSTED(NO) UACC(NONE)

Example 9

Operation User ADM1 wants to define the following:

- A SystemView for the MVS application named APPL1.HOST1.USER1
- TSOR220 application data
- A list of scripts named APPL1SC for the application
- A list of parameters named APPL1P for the application

Known User ADM1 has CLAUTH authority for the SYSMVIEW class.

Command RDEFINE SYSMVIEW APPL1.HOST1.USER1 APPLDATA('TSOR220')
SVFMR(SCRIPTNAME(APPL1SC) PARMNAME(APPL1P))

Defaults UACC(NONE)

Example 10

Operation Local realm KRB2000.IBM.COM is being defined with a minimum ticket lifetime of 5 minutes, a default ticket lifetime of 10 hours, a maximum ticket lifetime of 24 hours, and a password of 744275. All of the ticket lifetime values are specified in seconds.

Known The administrator has access to the KERBDFLT profile in the REALM class.

Command RDEFINE REALM KERBDFLT KERB(KERBNAME(KRB2000.IBM.COM)
MINTKTLFE(300) + DEFTKTLFE(36000) MAXTKTLFE(86400) PASSWORD(744275))

Defaults None

Example 11

Operation A trust relationship is being defined between the kerb390.endicott.ibm.com realm and the realm at ker2000.endicott.ibm.com.

Known The administrator has access to the
/.../KERB390.ENDICOTT.IBM.COM/KERBTGT/KER2000.ENDICOTT.IBM.COM
profile in the REALM class.

Command RDEFINE REALM
/.../KERB390.ENDICOTT.IBM.COM/KRBTGT/KER2000.ENDICOTT.IBM.COM
KERB(PASSWORD(12345678))

Defaults None

RDEFINE

Table 55. RDEFINE Examples (continued)

Example 12	<p><i>Operation</i> The administrator wants to create a profile (TSOIM13) in the PTKTDATA class with replay protection bypassed.</p> <p><i>Known</i> The administrator has the SPECIAL attribute.</p> <p><i>Command</i> RDEFINE PTKTDATA TSOIM13 APPLDATA('NO REPLAY PROTECTION')</p> <p><i>Defaults</i> None</p>
Example 13	<p><i>Operation</i> The administrator is defining the system wide defaults for Enterprise Identity Mapping (EIM) applications. One of the applications uses the default name given to RACF.</p> <p><i>Known</i> The EIM domain's distinguished name is 'ibm-eimDomainName=Pok EIM Domain,o=IBM,c=US'. The domain resides in ldap at 'http://some.big.host/'. The bind distinguished name has authority to retrieve lookup information. The name given to the local RACF registry is "RACFSYS2".</p> <p><i>Command</i> RDEFINE FACILITY IRR.PROXY.DEFAULTS EIM(DOMAINDN('ibm-eimDomainName=Pok EIM Domain,o=IBM,c=US') OPTIONS(ENABLE) LOCALREGISTRY(RACFSYS2))</p> <p><i>Defaults</i> None</p>
Example 14	<p><i>Operation</i> The administrator wants to define the SAFDFLT profile in the REALM class using the APPLDATA field to define the RACF realm name.</p> <p><i>Known</i> The administrator has the SPECIAL attribute. The realm name of racf.winmvs2c is selected by the security administrator to give a name to the set of user ids and other user information held in the security manager database. If Kerberos is in use in the installation, the Kerberos realm name would be expected to be different than the SAFDFLT realm name.</p> <p><i>Command</i> RDEFINE REALM SAFDFLT APPLDATA('racf.winmvs2c')</p> <p><i>Defaults</i> None</p>

RDELETE (Delete General Resource Profile)

Purpose

Use the RDELETE command to delete RACF resources belonging to classes specified in the class descriptor table.

This command removes the profile for the resource from the RACF database.

To have changes take effect after deleting a generic profile, if the class is not RACLISTed by either the SETROPTS RACLIST or RACROUTE REQUEST=LIST,GLOBAL=YES, one of the following steps is required:

- The security administrator issues the SETROPTS command:

```
SETROPTS GENERIC(class-name) REFRESH
```

See the SETROPTS command for authorization requirements.

- The user of the resource logs off and logs on again.

To have changes take effect after deleting a generic profile if the class is RACLISTed, the security administrator issues the following command:

```
SETROPTS RACLIST(class-name) REFRESH
```

For more information, refer to *z/OS Security Server RACF Security Administrator's Guide*.

Issuing Options

The following table identifies the eligible options for issuing the RDELETE command:

Table 56. How the RDELETE Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	Yes	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To delete a user profile, see “DELUSER (Delete User Profile)” on page 189.
- To delete a group profile, see “DELGROUP (Delete Group Profile)” on page 186.
- To delete a data set profile, see “DELDSD (Delete Data Set Profile)” on page 181.
- To obtain a list of general resource profiles, see “SEARCH (Search RACF Database)” on page 408.

RDELETE

Authorization Required

When issuing the RDELETE command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

To remove RACF protection from a resource in a class specified in the class descriptor table, you must have sufficient authority over the resource, so that one of the following conditions is met:

- You have the SPECIAL attribute.
- The resource profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You are the owner of the resource.
- If the profile is in the FILE or DIRECTRY class, the second qualifier of the profile name is your user ID.

For discrete profiles only:

- You are on the access list for the resource and you have ALTER authority. (If you have any other level of authority, you cannot use the command for this resource.)
- Your current connect group is on the access list and has ALTER authority. (If your group has any other level of authority, you cannot use the command for this resource.)
- The universal access authority for the resource is ALTER.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the RDELETE command is:

```
[subsystem-prefix]{RDELETE | RDEL}  
  
      class-name  
      (profile-name ...)  
      [ AT([node].userid ...) | ONLYAT([node].userid ...)]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

class-name

specifies the name of the class to which the resource belongs. Valid class names are those specified in the class descriptor table. For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, "Description of RACF Classes" on page 507.

This operand is required and must be the first operand following RDELETE.

This command is not intended to be used for profiles in the following classes:

- DCEUUIDS
- DIGTCERT
- DIGTRING
- NDSLINK
- NOTELINK
- TMEADMIN
- UNIXMAP

(profile-name...)

specifies the name of the existing discrete or generic profile RACF is to delete from the specified class. RACF deletes the profile for any resource you name by deleting it from the RACF database. RACF uses the class descriptor table to determine if the class is defined to RACF, the syntax of resource names within the class, and whether the resource is a group.

This operand is required and must be the second operand following RDELETE.

If you specify more than one value for *profile-name*, you must enclose the list of names in parentheses.

Mixed case profile names are accepted and preserved when *class-name* refers to a class defined in the class descriptor table with CASE=ASIS.

If you specify the *class-name* as CACHECLS, *profile-name* can either be *ccachname* or *cachname*.

Profiles in the CACHECLS hold the contents of a cache in profiles each containing 50K pieces of the cache. The profiles are named *cachename_001_00001*, *cachename_001_00002* and so forth, for as many profiles as are needed to hold the contents of the cache, where *cachename* was the Cache_name given as input on the R_cacheserv callable service. RDELETE for these profiles should only be used to correct an error condition, and it is expected that they will be used in response to an IRRL100xl message that was issued in response to invocation of the R_cacheserv SAF callable service. If for some reason, you want to delete the entire cache contents (perhaps because you do not want the contents used for authorization right

RDELETE

after an IPL), you can delete all of the *_ddd_nnnnn* profiles as well as the base profile by specifying just the *cachename* on the RDELETE.

If you specify *class-name* as a resource grouping class, you cannot specify a generic profile.

Notes:

1. If the resource you specify is a tape volume serial number that is a member of a tape volume set, RACF deletes the definitions for all of the volumes in the set.
2. RACF processes each resource you specify independently. If an error occurs while it is processing a resource, RACF issues a message and continues processing with the next resource.
3. You can use RDELETE to remove the profiles for a class defined to RACGLIST. For example, RDELETE RACGLIST TCICSTRN would remove the TCICSTRN base profile and any RACF-created TCICSTRN_XXXXXX profiles from the RACGLIST class. If you want to stop using RACGLIST for a particular class, issue the command RDELETE RACGLIST *class-name*. Do not delete specific RACF-created profiles unless RDELETE RACGLIST *class-name* was issued and failed to remove the profiles.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([*node*].*userid* ...)

specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT([*node*].*userid* ...)

specifies that the command is to be directed only to the node specified by *node* where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed only to the local node.

Examples

Table 57. RDELETE Examples

Example 1	<i>Operation</i>	User ADM2 wants to remove RACF protection from the terminals protected by the generic profile TERM*.
	<i>Known</i>	User ADM2 has the SPECIAL attribute. User ADM2 wants to issue the command as a RACF TSO command.
	<i>Command</i>	RDELETE TERMINAL TERM*
Example 2	<i>Defaults</i>	None
	<i>Operation</i>	User JHT01 wants to remove RACF protection from the tape volume set VOL001.
	<i>Known</i>	User JHT01 has the SPECIAL attribute. User JHT01 wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.
	<i>Command</i>	@RDELETE TAPEVOL VOL001
	<i>Defaults</i>	None

Table 57. RDELETE Examples (continued)

Example 3	<i>Operation</i>	User ADM1 wants to remove the generic profile T* from the TIMS class.
	<i>Known</i>	User ADM1 has the SPECIAL attribute. User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	RDELETE TIMS T*
	<i>Defaults</i>	None
Example 4	<i>Operation</i>	User ADM1 wants to delete the TERMINAL profiles in the RACGLIST class from the RACF database and stop using RACGLIST processing with the TERMINAL class. User ADM1 wants to direct the command to run at the node MVSFL under the authority of user JCARTER and prohibit the command from being automatically directed to other nodes.
	<i>Known</i>	Users ADM1 and JCARTER at MVSFL have the SPECIAL attribute. Users ADM1 and JCARTER at MVSFL have an already established user ID association. User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	RDELETE RACGLIST TERMINAL ONLYAT(MVSFL.JCARTER)
	<i>Results</i>	The command is only run at node MVSFL and not automatically directed to any other nodes in the RRSF configuration.

REMOVE

REMOVE (Remove User from Group)

Purpose

You can use the REMOVE command to remove a user from a group, and to assign a new owner to any group data set profiles the user owns on behalf of that group.

Issuing Options

The following table identifies the eligible options for issuing the REMOVE command:

Table 58. How the REMOVE Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	Yes	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To add a group profile, see “ADDGROUP (Add Group Profile)” on page 24.
- To change a group profile, see “ALTGROUP (Alter Group Profile)” on page 104.
- To connect a user to a group, see “CONNECT (Connect User to Group)” on page 173.
- To delete a group profile, see “DELGROUP (Delete Group Profile)” on page 186.
- To list a group profile, see “LISTGRP (List Group Profile)” on page 214.
- To display information from a user profile, see “LISTUSER (List User Profile)” on page 223.

Authorization Required

When issuing the REMOVE command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

To use the REMOVE command, one of the following conditions must be true:

- You have the SPECIAL attribute.
- The group profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You are the owner of the group.
- You have JOIN or CONNECT authority in the group.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

Notes:

1. If you only have ownership of the user's profile, you do not have sufficient authority to remove the user from a group.
2. If a user is deleted from a RACF group as a result of a REMOVE command while the user is logged on, the user must logoff and logon again before that authority to access resources in classes that have been RACLISTed is revoked. In addition, started tasks have to STOP and START to revoke the authority. This might include started tasks such as JES2 or JES3.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the REMOVE command is:

```
[subsystem-prefix]{REMOVE | RE}
                        (userid ...)
                        [ AT([node].userid ...) | ONLYAT([node].userid ...)]
                        [ GROUP(group-name) ]
                        [ OWNER(userid or group-name) ]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, "RACF TSO commands" on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, "RACF operator commands" on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

userid

specifies the user you want to remove from the group. If you are removing more than one user from the group, you must enclose the list of user IDs in parentheses.

This value is required and must be the first operand following REMOVE.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

REMOVE

AT([*node*].*userid* ...)

specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT([*node*].*userid* ...)

specifies that the command is to be directed only to the node specified by *node* where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed only to the local node.

GROUP(*group-name*)

specifies the group from which the user is to be removed. If you omit this operand, the default is your current connect group. The value specified for *group-name* cannot be the name of user's default group.

OWNER(*userid* or *group-name*)

specifies a RACF-defined user or group that owns the group data set profiles now owned by the user to be removed.

If you omit this operand when group data set profiles exist that require a new owner, RACF does not remove the user from the group. (Group data set profiles are data set profiles whose names are qualified by the group name or begin with the value supplied by an installation exit.)

The new owner of the group data set profiles must have at least USE authority in the specified group. Do not specify a user who is being removed from the group as the new data set profile owner.

Examples

Table 59. REMOVE Examples

Example 1

Operation User SIVLE wants to remove users KURT and JIMI from group PAYROLL.

Known User SIVLE has JOIN authority to group PAYROLL.

User SIVLE is currently connected to group PAYROLL.

Users KURT and JIMI are connected to group PAYROLL but do not own any group data set profiles, and group PAYROLL is not their default group.

User SIVLE wants to issue the command as a RACF TSO command.

Command REMOVE (KURT JIMI)

Defaults GROUP(PAYROLL)

Example 2

Operation User WRH0 wants to remove user PDJ6 from group RESEARCH, assigning user DAF0 as the new owner of PDJ6's group data set profiles.

Known User WRH0 has CONNECT authority to group RESEARCH.

User WRH0 is not logged on to group RESEARCH.

User PDJ6 is connected to group RESEARCH and owns group data set profiles (PDJ6's default connect group is not RESEARCH).

User DAF0 is connected to group RESEARCH with USE authority.

User WRH0 wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.

Command @REMOVE PDJ6 GROUP(RESEARCH) OWNER(DAF0)

Defaults None

RESTART (Restart RRSF Functions)

Purpose

Use the RESTART command to restart a function in the RACF subsystem address space. The RESTART command can be used after you apply maintenance and to recover from failures.

The RESTART command ends the current subtask and starts a new one. Only one function can be restarted with a single RESTART command, but that function might involve multiple subtasks.

Note: All users or applications that update the RACF database should be completed before issuing the RESTART command.

Issuing Options

The following table identifies the eligible options for issuing the RESTART command:

Table 60. How the RESTART Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
No	Yes	No	No	No

Note: For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To stop the RACF subsystem address space, see “STOP (Shutdown RRSF)” on page 479.
- To restart the RACF subsystem address space after it has been stopped, use the MVS START command.

Authorization Required

When issuing the RESTART command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the RESTART command is:

RESTART

subsystem-prefix RESTART

```
{COMMAND
 | CONNECTION [ NODE (nodename | *)
 | [ SYSNAME (sysname | *) ] ]
 | MESSAGE
 | OUTPUT
 | RACLINK
 | RECEIVE
 | SEND
 }
```

Note: For additional information on issuing this command as a RACF operator command, refer to “Rules for entering RACF operator commands” on page 21

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Subsystem prefix is a required keyword for RACF operator commands.

COMMAND | CONNECTION | MESSAGE | RACLINK | RECEIVE | SEND

specifies the restartable function.

Function Keyword	Module Name	Function
COMMAND	IRRSSC00	Command handler
CONNECTION	IRRDDM00	Device driver manager
	IRRAPPC0	Device drivers
	IRRAPPC2	
	IRRSSL00	Local node device driver
	IRRAPPC6	Handshaking
	IRRSSMG0	Message processor
MESSAGE	IRRSSMG0	
OUTPUT	IRRSSOP0	Output Handler
RACLINK	IRRSSK00	RACLINK task
RECEIVE	IRRSSR00	RRSF request receiver
SEND	IRRSSND0	RRSF request sender

Only one function keyword can be specified on the RESTART command at a time. If multiple function keywords are entered, the first one is processed and the rest is ignored.

The RESTART command signals the function specified by the function keyword to end and then restart.

NODE (*nodename* | *)

The NODE operand only applies to the CONNECTION keyword. If NODE is specified for a function other than CONNECTION, it is ignored.

Only a single RRSF nodename or * must be provided. Specifying * causes all device drivers to be ended and restarted. If a single nodename is provided, only the device drivers for that node are restarted.

If the **NODE** keyword is not specified with **CONNECTION**, the entire APPC server task, which includes all device drivers, is ended and restarted. As part of this processing, the device driver manager, device driver, and handshaking modules are reloaded: IRRDDM00, IRRAPPC0, IRRAPPC2, IRRAPPC6, and IRRSSL00. When the **NODE** keyword is specified, the device driver is ended and restarted, but no modules are reloaded.

SYSNAME(sysname / *)

restarts the connection to the specific member system *sysname* on the multisystem node *nodename*. The **SYSNAME** keyword is positional and must follow the **NODE** keyword. Specifying ***** causes all device drivers for each system in the particular node to end and restart.

Examples

Table 61. RESTART Examples

Example 1	<p><i>Operation</i> A maintenance PTF has been applied to module IRRSSC01 and user DNP2 wants to load a new copy to put the change in effect.</p> <p><i>Known</i> Module IRRSSC01 resides in load module IRRSSC00.</p> <p>The RACF subsystem prefix is @.</p> <p><i>Command</i> @RESTART COMMAND</p> <p><i>Defaults</i> None</p> <p><i>Results</i> The command handler is shut down and restarted. A new copy of load module IRRSSC00 is loaded including the updated copy of IRRSSC01.</p>
Example 2	<p><i>Operation</i> Restart the connection to the single-system node, NODE1</p> <p><i>Known</i> NODE1 is a single-system node. If it were not a single-system node, RACF would issue an error message and not execute the command.</p> <p><i>Command</i> RESTART CONNECTION NODE (NODE1)</p> <p><i>Defaults</i> None</p> <p><i>Results</i> The command restarts the connection to NODE1</p>
Example 3	<p><i>Operation</i> Restart the connections to all single-system nodes, and to all member systems of multisystem nodes</p> <p><i>Command</i> RESTART CONNECTION NODE (*) or RESTART CONNECTION NODE (*) SYSNAME(*)</p> <p><i>Defaults</i> None</p> <p><i>Results</i> The command restarts the connection to all single-system nodes and to all member systems of multisystem nodes.</p>
Example 4	<p><i>Operation</i> Restart the connections to the specific member system SYS1 on the multisystem node MULTNODE.</p> <p><i>Command</i> RESTART CONNECTION NODE (MULTNODE) SYSNAME(SYS1)</p> <p><i>Defaults</i> None</p> <p><i>Results</i> The command restarts the connection to SYS1 on MULTNODE. If MULTNODE is a single-system node, RACF issues an error message.</p>
Example 5	<p><i>Operation</i> Restart the connections to all member systems of the multisystem node MULTNODE.</p> <p><i>Command</i> RESTART CONNECTION NODE (MULTNODE) SYSNAME(*)</p> <p><i>Defaults</i> None</p> <p><i>Results</i> The command restarts the connection to all member systems of MULTNODE. If MULTNODE is a single-system node, RACF issues an error message.</p>

RLIST (List General Resource Profile)

Purpose

Use the RLIST command to display information on resources belonging to classes specified in the class descriptor table. Note that the DATASET, USER, and GROUP classes are not defined in the class descriptor table.

Note: The RLIST command might provide unpredictable results when searching on the DIGTCERT and DIGTRING classes. Due to the lower case characters in these classes, the profile filter on the RLIST command might not function correctly.

RACF uses the class descriptor table to determine if a class is defined to RACF, the syntax of resource names within the class, and whether the class is a resource grouping class.

Profiles are listed in alphabetical order. Generic profiles are listed in the same order as they are searched for a resource match. (This also applies to the names in the global access table.)

Note: RACF interprets dates with 2 digit years in the following way. YY represents the 2 digit year.

```
IF 70 < YY <= 99 THEN
```

The date is interpreted as 19YY

```
IF 00 <= YY <= 70 THEN
```

The date is interpreted as 20YY

Issuing Options

The following table identifies the eligible options for issuing the RLIST command:

Table 62. How the RLIST Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	No	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To list a data set profile, see “LISTDSD (List Data Set Profile)” on page 200.
- To list a user profile, see “LISTUSER (List User Profile)” on page 223.
- To list a group profile, see “LISTGRP (List Group Profile)” on page 214.
- To obtain a list of general resource profiles, see “SEARCH (Search RACF Database)” on page 408.

Comments

This command lists the information in an existing profile for the resource or resource group.

The details that are given for each profile are:

- The resource class.
- The name of the resource.
- The cross-reference class name (that is, the member class name for resource groups or the group name for non-group resources).
- If the resource named in the command (in the resource-name operand) is a resource group, RACF lists member resources.
- The level of the resource.
- The owner of the resource.
- The type of access attempts (as specified by the AUDIT operand on the RDEFINE or RALTER command) that are being logged on the SMF data set.
- The user, if any, to be notified when RACF uses this profile to deny access to the resource.
- The universal access authority for the resource.
- Your highest level of access authority to the resource.
- The installation-defined data (information specified in the DATA operand of the RALTER or RDEFINE commands).

If your installation is configured to be a B1 security environment, this information is not listed in your output. * SUPPRESSED * appears under the installation data field. Only those with SPECIAL are allowed to list the field.

- The APPLDATA value, if any.

If your installation is configured to be a B1 security environment, this information is not listed in your output. * SUPPRESSED * appears under the application data field. Only those with SPECIAL are allowed to list the field.

- The domain distinguished name, options and local registry for the EIM segment.
- The type of access attempts (as specified by the GLOBALAUDIT operand on the RALTER command) that RACF logs.
- The status of the WARNING/NOWARNING indicator.
- For resources in the TAPEVOL class:
 - The volumes in a tape volume set,
 - Whether the TAPEVOL profile is automatic or nonautomatic,
 - Whether the volume can hold more than one data set, or
 - Whether the volume contains a TVTOC.

Additional details:

You can request the following details by using the appropriate RLIST operands:

- The security label, the security level and categories.
(See the AUTHUSER operand.)
- For member resources, RACF lists the names of all resource group members in which the entity is a member.
(See the RESGROUP operand.)
- The number of times the resource was accessed by all users for each of the following access authorities.

RLIST

- ALTER, CONTROL, UPDATE, READ

(See the STATISTICS operand. This detail is only meaningful when your installation is gathering resource statistics and the class is not RACLISTed. For a generic profile, RACF replaces any statistics line with NOT APPLICABLE FOR GENERIC PROFILE.)

- Historical data, such as:
 - Date the resource was defined to RACF,
 - Date the resource was last referenced (this detail is only meaningful when your installation is gathering resource statistics and the class is not RACLISTed; for a generic profile, RACF replaces any statistics line with NOT APPLICABLE FOR GENERIC PROFILE), or
 - Date the resource was last accessed at the update level.

(See the HISTORY operand.)

- The standard access list which displays:
 - All users and groups authorized to access the resource,
 - The level of authority for each user and group, or
 - The number of times each user has accessed the resource. (This detail is only meaningful when your installation is gathering resource statistics. This detail is not included in the output for generic profiles.)

(See the AUTHUSER operand.)

- The conditional access list which displays the same fields as the standard access list, as well as the following additional fields:
 - The class of the resource, or
 - The entity name of the resource.

(See the AUTHUSER operand.)

- For a tape volume that contains RACF-protected data sets, the following information about each RACF-protected data set on the volume:
 - The name used to create the data set,
 - The internal RACF name for the data set,
 - The volumes on which the data set resides,
 - The file sequence number for the data set,
 - The date when the data set was created, or
 - Whether the data set profile is discrete or generic.

(See the TVTOC operand.)

- The contents of segments other than the base segment, such as DLFDATA.
(See the segment operands for details on what is listed.)
- The contents of the SESSION segment.
(See the SESSION operand.)
- The contents of the SSIGNON segment.
(See the SSIGNON operand.)
- The contents of the STDATA segment.
(See the STDATA operand.)
- The contents of the TME segment.
(See the TME operand.)

Authorization Required

When issuing the RLIST command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

You must have a sufficient level of authority for each resource or resource group listed as the result of your request so that one of the following conditions is met:

- You have the SPECIAL attribute.
- The resource profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You have the OPERATIONS attribute.
- The resource profile is within the scope of a group in which you have the group-OPERATIONS attribute.
- You have the AUDITOR attribute
- The resource profile is within the scope of a group in which you have the group-AUDITOR attribute.
- You are the owner of the resource.
- If the profile is in the FILE or DIRECTRY class, the second qualifier of the profile name is your user ID.
- To list the contents of segments other than the base segment, such as DLFDATA, SPECIAL, AUDITOR, or field-level access checking is required.
- You are on the access list for the resource and you have at least READ authority. (If your level of authority is NONE, the resource is not listed.) If you specify ALL, RACF lists only information pertinent to your user ID.
- Your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is in the access list and has at least READ authority.
- The universal access authority of the resource is at least READ.
- You have at least read access for the profile name from the GLOBAL ENTRY TABLE (if this table contains an entry for the profile).

You see the type of access attempts, as specified by the GLOBALAUDIT operand, only if you have the AUDITOR attribute or if the resource profile is within the scope of a group in which you have the group-AUDITOR attribute.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

Listing Resource Access Lists: When you are requesting to see the access list for a resource with the AUTHUSER operand, your level of authority is checked for each resource. Your level of authority must be such that one of the following conditions is met:

- You have the SPECIAL attribute.
- The resource profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You have the OPERATIONS attribute.

RLIST

- The resource profile is within the scope of a group in which you have the group-OPERATIONS attribute.
- You are the owner of the resource.
- You have the AUDITOR attribute.
- The resource profile is within the scope of a group in which you have the group-AUDITOR attribute.
- You have alter access for the profile name from the GLOBAL ENTRY TABLE (if this table contains an entry for the profile).
- You are on the access list for the resource and you have ALTER authority. (If you have any other level of authority, you can not use the operand.)
- Your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is in the access list and has ALTER authority. (If any group that RACF checked has any other level of authority, you can not use the operand.)
- The universal access authority of the resource is ALTER.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the RLIST command is:

```
[subsystem-prefix]{RLIST | RL}
                        class-name
                        {(profile-name ...) | * }
                        [ ALL ]
                        [ AT([node].userid ...) | ONLYAT([node].userid ...) ]
                        [ AUTHUSER ]
                        [ DLFDATA ]
                        [ EIM ]
                        [ {GENERIC | NOGENERIC} ]
                        [ HISTORY ]
                        [ KERB ]
                        [ NORACF ]
                        [ NOYOURACC ]
                        [ PROXY ]
                        [ RESGROUP ]
                        [ SESSION ]
                        [ SSIGNON ]
                        [ STATISTICS ]
                        [ STDATA ]
                        [ SVFMR ]
                        [ TME ]
                        [ TVTOC ]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

class-name

specifies the name of the class to which the resource belongs. Valid class names are those specified in the class descriptor table. For a list of general resource classes supplied by IBM, see Appendix B, "Description of RACF Classes" on page 507.

This operand is required and must be the first operand following RLIST.

If you specify a class that allows resource names to contain any character (ICHERCDE with OTHER=ANY), resources in the DATASET class that begin with the same first four characters as this class name are also be listed.

(profile-name ...) | *

(profile-name...)

Specifies the name of an existing discrete or generic profile about which information is to be displayed.

The variable *profile-name* or an asterisk (*) is required and must be the second operand following RLIST.

If you specify more than one value for *profile-name*, the list of names must be enclosed in parentheses.

Mixed case profile names are accepted and preserved when *class-name* refers to a class defined in the class descriptor table with CASE=ASIS.

If the resource specified is a tape volume serial number that is a member of a tape volume set, information on all the volumes in the set are displayed.

RACF processes each resource you specify independently. If an error occurs while processing a resource, RACF issues a message and continues processing with the next resource.

- * Specifies that you want to display information for all resources defined to the specified class for which you have the proper authority.

On a system with many profiles defined, the use of * may result in a large amount of output that may not be useful to a user issuing the command. It may be more appropriate for the user to browse the output of IRRDBU00 (database unload) or to write a program to process the IRRDBU00 output and produce a report showing only the subset of information that is of interest to the user. The processing of output of RLIST by programs is not supported nor recommended by IBM. If you want a listing of all the profiles for use by a program you should instead have the program process the output from IRRDBU00, RACROUTE REQUEST=EXTRACT, or ICHEINTY.

An asterisk (*) or *profile-name* is required and must be the second operand following RLIST.

RLIST

RACF processes each resource independently and displays information only for those resources for which you have sufficient authority.

If you have the AUDITOR attribute, or if the resource profile is within the scope of a group in which you have the group-AUDITOR attribute, RACF displays GLOBALAUDIT information for all resources in the class.

ALL

specifies that you want all information for the BASE segment of each resource displayed.

The access list is included only if you have sufficient authority to use the AUTHUSER operand (see Authorization Required). The type of access attempts (as specified by the GLOBALAUDIT operand) that are being logged on the SMF data set is included only if you have the AUDITOR attribute, or the resource profile is within the scope of a group in which you have the group-AUDITOR attribute.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([*node*].*userid* ...)

specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT([*node*].*userid* ...)

RLIST is not eligible for automatic command direction. If you specify the ONLYAT keyword, the effect is the same as if you specified the AT keyword.

AUTHUSER

specifies that you want the following information included in the output:

- The user categories authorized to access the resource
- The security level required to access the resource
- The security label required to access the resource
- The standard access list. This includes the following:
 - All users and groups authorized to access the resource
 - The level of authority for each user and group
 - The number of times the user has accessed the resource (This detail is only meaningful when your installation is gathering resource statistics and is not included in the output for generic profiles.)
- The conditional access list. This list consists of the same fields as in the standard access list, as well as the following fields:
 - The class of the resource through which each user and group in the list can access the target resource of the command. For example, if a user can access the target resource through terminal TERM01, then TERMINAL would be the class listed.
 - The entity name of the resource through which each user and group in the list can access the target resource of the command. In the example above, TERM01 would be listed.

You must have sufficient authorization to use the AUTHUSER operand (see Authorization Required).

DLFDATA

list the contents of the DLFDATA segment for profiles in the DLFCLASS class.

EIM

Specifies that EIM segment information should be listed.

GENERIC | NOGENERIC**GENERIC**

specifies that you want RACF to display information for the generic profile that most closely matches a resource name. If you specify **GENERIC**, RACF ignores a discrete profile that protects the resource. If asterisk (*) is specified instead of the profile name, all generic profiles are listed.

NOGENERIC

specifies that you want RACF to display information for the discrete profile that protects a resource. If asterisk (*) is specified instead of the profile name, all discrete profiles are listed.

If neither **GENERIC** nor **NOGENERIC** is specified, RACF lists information for the discrete resource name that matches the resource name you specify. If there is no matching discrete profile, RACF lists the generic profile that most closely matches the resource name. If asterisk (*) is specified instead of the profile name, all discrete and generic profiles are listed.

The following list shows examples of using the **GENERIC** and **NOGENERIC** operands:

- If you enter:

```
RLIST DASDVOL *
```

RACF lists all discrete and generic profiles in the DASDVOL class.

- If you enter:

```
RLIST DASDVOL * GENERIC
```

RACF lists information for all the generic profiles in the DASDVOL class.

- If you enter:

```
RLIST JESSPOOL * NOGENERIC
```

RACF lists all discrete profiles in the JESSPOOL class.

- If you enter:

```
RLIST APPCLU ABC.DEF GENERIC
```

RACF displays the best-fit generic profile that protects the resource ABC.DEF. RACF ignores discrete profile ABC.DEF if it exists.

Note: When searching for a generic profile that matches the specified resource, RACF does not examine members that are defined in a grouping class (through the **ADDMEM** operand of the **RDEFINE** command). For example, suppose two profiles had been defined by the following **RDEFINE** commands:

```
RDEFINE TCICSTRN A*
RDEFINE GCICSTRN xxx ADDMEM(AB*)
```

The command:

```
RLIST TCICSTRN ABC
```

RLIST

displays profile A* in the TCICSTRN class, but it does not search the GCICSTRN class and therefore does not display any AB* profile of the GCICSTRN class. In addition, the command:

```
RLIST GCICSTRN ABC
```

does not find member AB* in the GCICSTRN class because it does not look at the members in a grouping class.

If you wish to make use of RLIST to find the generic profile that protects a specific resource, and the resource is in a class that has both a grouping class and a member class, you should define the generic profile as a profile in the member class.

To illustrate the above RDEFINE example where ADDMEM(AB*) had been specified for a grouping class, the following command:

```
RDEFINE TCICSTRN AB*
```

allows the RLIST command to display AB* as the generic member in the TCICSTRN class.

HISTORY

specifies that you want to list the following data:

- The date each profile was defined to RACF
- The date each profile was last referenced (this detail is only meaningful when your installation is gathering resource statistics; for a generic profile and profiles that are RACLISTed, RACF replaces any statistics line with NOT APPLICABLE FOR GENERIC PROFILE)
- The date of last RACROUTE REQUEST=AUTH for UPDATE authority (this detail is only meaningful when your installation is gathering resource statistics; for a generic profile and profiles that are RACLISTed, RACF replaces any statistics line with NOT APPLICABLE FOR GENERIC PROFILE)

KERB

specifies that you want to list the following Security Server Network Authentication Service information:

- The local *kerberos-realm-name* (KERBNAME)
- The encryption value settings, for example, ENCRYPT(DES DES3 DESD) or NOENCRYPT
- The *min-ticket-life* value for the local realm (MINTKTLFE)
- The *def-ticket-life* value for the local realm (DEFTKTLFE)
- The *max-ticket-life* value for the local realm (MAXTKTLFE)
- The current key version (KEY VERSION)

Note: If KEY VERSION is not displayed, there is no Network Authentication Service key associated with this realm definition.

NORACF

specifies that you want to suppress the listing of RACF segment information. If you specify NORACF, you must include either DLFDATA, KERB, PROXY, SESSION, SSIGNON, STDATA, TME, or a combination of operands.

If you do not specify NORACF, RACF displays the information in the base segment of a general resource profile.

The information displayed as a result of using the NORACF operand is dependent on other operands used in the command. For example, if you use NORACF with SESSION also specified, only the SESSION information is displayed.

NOYOURACC

For grouping and member classes, RLIST must do additional processing to assure that the *your access* information field is accurate. A SPECIAL user can use the NOYOURACC operand to bypass this processing, for performance reasons. The *your access* field contains “n/a” in this circumstance.

Note: This operand applies to SPECIAL users only. It has no effect for other users.

PROXY

Specifies that PROXY segment information should be listed. The following information will be provided:

- the URL of the LDAP server to be contacted
- the BIND distinguished name
- information regarding the BIND password

The BINDPW password values will not be listed. If a BINDPW password value is defined for a general resource profile, RLIST will display 'YES' for the PROXY segment BINDPW attribute. If no BINDPW password value has been defined, RLIST will display 'NO' for the PROXY segment BINDPW attribute.

RESGROUP

requests a list of all resource groups of which the resource specified by the *profile-name* operand is a member.

If a profile does *not* exist for the specified resource, RACF lists the names of all resource groups of which the resource is a member and to which the command user is authorized. To be authorized, the command user must meet one of the RACF requirements listed in Authorization Required.

If a profile *does* exist for the specified resource and the command user has ALTER authority to the resource, RACF lists the names of all groups of which the resource is a member.

If a profile *does* exist for the specified resource but the command user has less than ALTER authority to the resource, RACF lists the names of all groups of which the resource is a member and to which the command user is authorized. To be authorized to the resource group, the command user must meet one of the RACF requirements listed in Authorization Required. However, the command issuer must have the authority to list the resource specified on the command in order to list the member groups. If this requirement is met, then the user must be also authorized to the resource group. Otherwise, an error message is issued.

When *profile-name* is the name of a protected resource (such as a terminal or DASD volume) and *class-name* is a “member class” (such as TERMINAL or DASDVOL), the RESGROUP operand lists the profiles that protect the resource (for example, profiles in the GTERMINL or GDASDVOL class).

If you define a profile and use generic characters such as (*) to add members to the profile, RLIST RESGROUP will not return any of the matching profiles in its output because it does not support generic matches. For example, you have:

```
RDEF GIMS GIMSGRP ADDMEM(ABC*)
```

RLIST

and you are looking for a specific member, so you enter:

```
RLIST GIMS ABCD RESGROUP
```

ABC* will not appear in the output.

Note: When considering this example, if you are unable to define the profile ABCD, it might be due to a generic definition somewhere in GIMS.

This operand applies only to “member classes” for which resource group profiles exist.

SESSION

specifies that the contents of the SESSION segment are to be listed for profiles in the APPCLU class.

SSIGNON

indicates that you want to display the secured signon information.

Note: The secured signon application key value cannot be displayed. However, information is displayed that describes whether the key value is masked or encrypted.

STATISTICS

specifies that you want to list the statistics for each resource. The list contains the number of times the resource was accessed by users with READ, UPDATE, CONTROL, and ALTER authorities. A separate total is given for each authority level.

Note: This detail is only meaningful when your installation is gathering resource statistics. For a generic profile, RACF replaces any statistics line with NOT APPLICABLE FOR GENERIC PROFILE.

STDATA

specifies that you want to list the contents of the STDATA segment for profiles in the STARTED class.

SVFMR

lists the contents of the SVFMR segment for profiles in the SYSMVIEW class.

TME

specifies that information in the Tivoli Security Management Application is to be listed.

TVTOC

specifies that you want to see information about the data sets defined in the TVTOC of a TAPEVOL profile. The output displays:

- The name used to create the data set
- The internal RACF name for the data set
- The volumes on which the data set resides
- The file sequence number for the data set
- The date when the data set was created
- Whether the data set profile is discrete or generic.

Examples

Table 63. RLIST Examples

Example 1	<i>Operation</i>	User RV2 wants to list all information about the tape volume VOL001.
	<i>Known</i>	User RV2 is the owner of tape volume VOL001.
		User RV2 has the AUDITOR attribute.
		User RV2 wants to issue the command as a RACF TSO command.
Example 2	<i>Command</i>	RLIST TAPEVOL VOL001 ALL
	<i>Defaults</i>	None
	<i>Output</i>	See Figure 43.
	<i>Operation</i>	User ADM1 wants to list information about the generic profile T* in the TIMS class.
Example 3	<i>Known</i>	User ADM1 has the SPECIAL and AUDITOR attributes.
		User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	RLIST TIMS T*
	<i>Defaults</i>	None
Example 4	<i>Output</i>	See Figure 44.
	<i>Operation</i>	User IBMUSER wants to list information about the profile TERM1 in the TERMINAL class. TERM1 is a member of four GTERMINL class profiles: GTERM1, GTERM2, GTERM3, and GTERM4. TERM1 has a UACC of NONE.
	<i>Known</i>	User IBMUSER has the SPECIAL and AUDITOR attributes. User IBMUSER wants to issue the command as a RACF TSO command.
	<i>Command</i>	RLIST TERMINAL TERM1 RESGROUP
Example 5	<i>Defaults</i>	None
	<i>Output</i>	See Figure 45.
	<i>Operation</i>	The security administrator wants to display secured signon key information for profile name TSOR001 in the PTKTDATA class to be certain that the application key is masked instead of encrypted.
	<i>Known</i>	SIVLE1 is the user ID of the security administrator and has the SPECIAL attribute. The security administrator wants to issue the command as a RACF TSO command.
Example 6	<i>Command</i>	RLIST PTKTDATA TSOR001 SSIGNON
	<i>Defaults</i>	None
	<i>Output</i>	See Figure 46.
	<i>Operation</i>	The security administrator wants to display secured signon key information for profile name TSOR004 in the PTKTDATA class and to be certain that the application key is encrypted instead of masked.
Example 7	<i>Known</i>	NONNEL is the user ID of the security administrator and has the SPECIAL attribute. The security administrator wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.
	<i>Command</i>	@RLIST PTKTDATA TSOR004 SSIGNON
	<i>Defaults</i>	None
	<i>Output</i>	See Figure 47.
Example 8	<i>Operation</i>	The security administrator wants to display the contents of the STDATA segments for profiles in the STARTED class with the generic profile name VTAM®.*.
	<i>Known</i>	SYSUSER is the user ID of the security administrator and has the SPECIAL attribute. The security administrator wants to issue the command as a RACF TSO command.
	<i>Command</i>	RLIST STARTED VTAM.* STDATA NORACF
	<i>Defaults</i>	None
Example 9	<i>Output</i>	See Figure 48.

RLIST

Table 63. RLIST Examples (continued)

Example 7	<i>Operation</i>	The security administrator wants to list the contents of the KERBDFLT profile in the REALM class.
	<i>Known</i>	The administrator has access to the KERBDFLT profile in the REALM class.
	<i>Command</i>	RLIST REALM KERBDFLT KERB NORACF
	<i>Defaults</i>	None
Example 8	<i>Output</i>	See Figure 49.
	<i>Operation</i>	The administrator wants to list the contents of a profile (TSOIM13) in the PTKTDATA class. This particular PassTicket profile indicates that replay protection is to be bypassed.
	<i>Known</i>	The administrator has access to the PTKTDATA class.
	<i>Command</i>	RLIST PTKTDATA TSOIM13
Example 9	<i>Defaults</i>	None
	<i>Output</i>	See Figure 50.
	<i>Operation</i>	The administrator wants to list the contents of a profile (IRR.PROXY.DEFAULTS) in the FACILITY class and the contents of the EIM segment. This particular PROXY profile indicates that a BINDPW has been defined.
	<i>Known</i>	The administrator has access to the FACILITY class.
	<i>Command</i>	RLIST FACILTY IRR.PROXY.DEFAULTS EIM PROXY NORACF
	<i>Defaults</i>	None
	<i>Output</i>	See Figure 51.

```

RLIST TAPEVOL VOL001 ALL
CLASS      NAME
-----
TAPEVOL    VOL001
LEVEL OWNER  UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----
  00   RV2      READ           ALTER        NO
INSTALLATION DATA
-----
NONE
APPLICATION DATA
-----
NONE
SECLEVEL
-----
NO SECLEVEL
CATEGORIES
-----
NO CATEGORIES
SECLABEL
-----
NO SECLABEL
AUDITING
-----
SUCCESS(READ), FAILURES(UPDATE)
GLOBALAUDIT
-----
ALL(CONTROL)
AUTOMATIC  SINGLE DATA SET
-----
      NO      NO
NOTIFY
-----
NO USER TO BE NOTIFIED
CREATION DATE  LAST REFERENCE DATE  LAST CHANGE DATE
(DAY) (YEAR)    (DAY) (YEAR)    (DAY) (YEAR)
-----
  146   82      146   82      146   82
ALTER COUNT  CONTROL COUNT  UPDATE COUNT  READ COUNT
-----
000000      000000      000005      000000
USER      ACCESS  ACCESS COUNT
-----
RV2      ALTER      000000
ESH25    READ      000000
ID      ACCESS  ACCESS COUNT  CLASS  ENTITY NAME
--      -----
NO ENTRIES IN CONDITIONAL ACCESS LIST
NO TVTOC INFORMATION AVAILABLE

```

Figure 43. Example 1: Output for the RLIST Command

RLIST

```

RLIST TIMS T*
CLASS      NAME
-----
TIMS      T* (G)
GROUP     CLASS  NAME
-----
GIMS
RESOURCE GROUPS
-----
NONE
LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----
00     ADM1      NONE              ALTER        NO
INSTALLATION DATA
-----
NONE
APPLICATION DATA
-----
REVERIFY
AUDITING
-----
NONE
GLOBALAUDIT
-----
SUCCESS(UPDATE), FAILURES(READ)
NOTIFY
-----
NO USER TO BE NOTIFIED

```

Figure 44. Example 2: Output for the RLIST Command

```

RLIST TERM1
CLASS      NAME
-----
TERMINAL  TERM1
GROUP     CLASS  NAME
-----
GTERMINL
RESOURCE GROUPS
-----
GTERM1 GTERM2 GTERM3 GTERM4
LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----
00     IBMUSER      NONE              ALTER        NO
INSTALLATION DATA
-----
NONE
APPLICATION DATA
-----
NONE
AUDITING
-----
FAILURES(READ)
TIMEZONE  LOGON ALLOWED  (DAYS)      (TIME)
-----
CPU TIME  ANYDAY      ANYTIME
NOTIFY
-----
NO USER TO BE NOTIFIED

```

Figure 45. Example 3: Output for the RLIST Command with RESGROUP Option

```
SSIGNON INFORMATION
-----
KEYMASKED DATA NOT DISPLAYABLE
```

Figure 46. Example 4: Output for RLIST Command with Masked Application Key

```
SSIGNON INFORMATION
-----
KEYENCRYPTED DATA NOT DISPLAYABLE
```

Figure 47. Example 5: Output for RLIST Command with Encrypted Application Key

```
STDATA INFORMATION
-----
USER= SYSUSER
GROUP= SYSGROUP
TRUSTED= YES
PRIVILEGED= NO
TRACE= NO
```

Figure 48. Example 6: Output for RLIST Command for STDATA Segment

```
CLASS      NAME
-----
REALM      KERBDFLT

KERB INFORMATION
-----
KERBNAME= KRB2000.IBM.COM
MINTKTLFE= 0000000300
MAXTKTLFE= 0000086400
DEFTKTLFE= 0000036000
KEY VERSION= 001
-----
CLASS      NAME
-----
REALM      /.../KERB390.ENDICOTT.IBM.COM/KRBTGT/KER2000.ENDICOTT.IBM.COM

...
```

Figure 49. Example 7: Output for RLIST Command for KERB Segment

RLIST

```

CLASS      NAME
-----
PTKTDATA   TSOIM13

LEVEL      OWNER    UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----
00         IBMUSER      NONE             NONE          NO

INSTALLATION DATA
-----
NONE

APPLICATION DATA
-----
NO REPLAY PROTECTION

AUDITING
-----
FAILURES(READ)

NOTIFY
-----
NO USER TO BE NOTIFIED
READY

```

Figure 50. Example 8: Output for RLIST Command for TSOIM13

```

RLIST FACILITY IRR.PROXY.DEFAULTS PROXY NORACF
CLASS      NAME
FACILITY    IRR.PROXY.DEFAULTS

PROXY INFORMATION
LDAPHOST= LDAP://SOME.LDAP.HOST:389
BINDDN= cn=Joe User,ou=Poughkeepsie,o=IBM,c=US
BINDPW= YES

```

Figure 51. Example 9: Output for RLIST FACILITY IRR.PROXY.DEFAULTS PROXY NORACF

```

RLIST FACILITY IRR.PROXY.DEFAULTS EIM PROXY NORACF
CLASS      NAME
FACILITY    IRR.PROXY.DEFAULTS

EIM INFORMATION
DOMAINDN= Ibm-eimDomainName=Pok EIM Domain,o=IBM,c=US
OPTIONS= ENABLE
LOCALREGISTRY= RACFSYS2

```

Figure 52. Example 10: Output for RLIST FACILITY IRR.PROXY.DEFAULTS EIM PROXY NORACF

RVARY (Change Status of RACF Database)

Purpose

Use the RVARY command to:

- Deactivate and reactivate the RACF function.
- Switch from using a specific primary data set to using its corresponding backup data set, perhaps because of a failure related to the primary data set.
- Deactivate or reactivate primary or backup RACF data sets. (Deactivating a specific primary data set causes all RACF requests for access to that data set to fail. Deactivating a specific backup data set causes RACF to stop duplicating information on that data set.)
- Deactivate protection for any resources belonging to classes defined in the class descriptor table while RACF is inactive.
- Select the mode of operation when RACF is enabled for sysplex communication.

While RACF is deactivated, utilities can be run to diagnose and repair logical errors in the RACF database. RACF installation exits can provide special handling for requests to access RACF-protected resources (for example, by prompting the operator to allow or deny access). If the RACF data set is itself RACF-protected, RACF failsoft processing, which can include installation exit routine processing, controls access to the RACF database. When you deactivate RACF using the RVARY command, only users defined in TSO SYS1.UADS can still log on to TSO, and RACF does not validate those user IDs. When RACF is inactive, failsoft processing takes effect.

Note: Failsoft processing occurs only when all primary RACF data sets are inactive. If you have multiple RACF data sets and only one is inactive, you are likely to experience ABENDs. See *z/OS Security Server RACF System Programmer's Guide* for more information on failsoft processing and using RVARY.

RACF logs each use of the RVARY command provided that the system has been IPLed with RACF active and the use of RVARY changes the status of RACF. For example, if you issue RVARY to deactivate a RACF database that is already inactive, you do not change the status of RACF. Therefore, RACF does not log this particular use of RVARY. When RACF is enabled for sysplex communication, logging of the RVARY commands occurs only on the system from which the command originated.

When you deactivate a RACF data set (using RVARY INACTIVE) or switch to a backup RACF data set (using RVARY SWITCH), RACF automatically deallocates that data set. To reactivate a data set, use the RVARY ACTIVE command. The RVARY SWITCH does not activate an inactive data set. RACF automatically reallocates that data set. This feature allows you to restore the data set from a copy on tape or recatalog the data set on another volume without having to re-IPL your system.

If you deactivate the primary RACF data set, and uncatalog it, and replace it with an alternate data set, the alternate data set must be cataloged and have the same name as the original data set before you can activate it. When you deactivate (and deallocate) a RACF data set, you can move the data set from one direct access storage device to another.

RVARY

Before recataloging a data set, you must first deactivate the data set by issuing either the RVARY INACTIVE or the RVARY SWITCH command.

Using RVARY when RACF is enabled for sysplex communication: In addition to the RVARY DATASHARE and RVARY NODATASHARE commands, which are valid *only* when RACF is enabled for sysplex communication, the following RVARY commands are propagated when RACF is enabled for sysplex communication:

- RVARY ACTIVE
- RVARY INACTIVE
- RVARY SWITCH

When issued from any member of the RACF data sharing group, these commands are propagated in a controlled, synchronized manner to each of the other members in the group.

Notes:

1. For RVARY INACTIVE(NOCLASSACT) and RVARY INACTIVE(NOTAPE) commands, only the RVARY INACTIVE portion of the command is propagated.
2. The MVS operator commands ROUTE *ALL and ROUTE *system-group-name* are allowed only with RVARY LIST.
3. RACF does not propagate commands if the system is operating in failsoft mode unless failsoft mode was entered because an RVARY INACTIVE command was issued.
4. RVARY INACTIVE DATASET, SWITCH, DATASHARE, and NODATASHARE require that RVARY quiesce RACF database I/O activity before proceeding. There can be no database I/O activity in progress while the status of the database is changed or the database could get corrupted. Consequently, RVARY must wait for previously scheduled database I/O to complete before proceeding. If there are problems with the DASD device the data set is on and the I/O is hung, those problems have to be cleared up before the command can complete. See the RVARY command documentation in the 'Recovery Procedures' chapter of the *z/OS Security Server RACF System Programmer's Guide* for more information.

Issuing Options

The following table identifies the eligible options for issuing the RVARY command:

Table 64. How the RVARY Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	No	No	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, "RACF TSO commands" on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, "RACF operator commands" on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Authorization Required

No special authority is needed to issue the RVARY command. However, the operator (at the operator console or security console) must approve a change in RACF status or the RACF data sets—or a change in the operational mode if RACF is enabled for sysplex communication—before RACF allows the command to complete.

If the RVARY command changes RACF or database status (ACTIVE/INACTIVE), RACF issues an informational message and the operator is required to enter the password defined by RVARYPW STATUS(*status-pw*) to authorize the change. If the RVARY command switches the RACF data sets (SWITCH) or changes the RACF operating mode (DATASHARE/NODATASHARE), RACF issues an informational message and the operator is required to enter the password defined by RVARYPW SWITCH(*switch-pw*). When RVARY is issued as a RACF operator command from a console with master authority, the default password YES is also accepted for RVARY ACTIVE, RVARY NODATASHARE or RVARY SWITCH commands.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the RVARY command is:

```
[subsystem-prefix]RVARY
                        [ ACTIVE | INACTIVE [ NOCLASSACT(class-namelist | *)
                          (NOTAPE) ]
                        | DATASHARE | NODATASHARE | SWITCH ]
                        [ DATASET(dataset-name... | *) ]
                        [ LIST | NOLIST ]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

ACTIVE | INACTIVE [NOCLASSACT (*class-namelist* | *) (NOTAPE)] | DATASHARE | NODATASHARE | SWITCH

ACTIVE

specifies that the RACF function for, and access to, the primary RACF database is to be reactivated.

RVARY

If you wish to reactivate a particular primary data set or if you wish to activate or reactivate a backup data set, then you must specify the DATASET operand with the appropriate data set name.

When you reactivate any RACF data set it is automatically reallocated.

When RACF is enabled for sysplex communication and RVARY ACTIVE is issued from one member of a sysplex data sharing group, RACF attempts to connect every peer system that is in data sharing mode or in read-only mode to the coupling facility structures associated with each active database. If any connection attempt fails, the affected member enters read-only mode, although the data set will remain allocated and active. The system originating the command might be in either data sharing mode or in read-only mode.

INACTIVE

specifies that the RACF function for, and access to, the RACF database is to be deactivated.

To deactivate a particular primary data set or a backup data set, specify the DATASET operand with the appropriate data set name. If the DATASET operand is not specified, the default is all primary RACF data sets.

If your installation did not specify a backup database in the data set name table (ICHRDSNT), and you need to deactivate the primary database, you must use the RVARY INACTIVE command. If you have only a single data set, your system enters failsoft processing. If you have multiple data sets and only some are active, you are likely to experience ABENDs.

When you deactivate any RACF data set, it is automatically deallocated.

If RACF is enabled for sysplex communication, RACF disconnect from any coupling-facility-related structures that are in use by members of a RACF sysplex data sharing group running in data sharing mode or in read-only mode.

If the data set specified in the RVARY INACTIVE command is associated with a coupling facility structure failure, or with a connection failure and there is no other failed structure or connection, the system can be put into data sharing mode as a result of RVARY INACTIVE.

INACTIVE NOCLASSACT(*class-namelist* | *)

specifies those classes for which RACF protection is not in effect while RACF is inactive. The variable *class-namelist* can contain any class defined in the class descriptor table, while '*' indicates that the operand applies to all classes defined in class descriptor table. This option takes effect immediately and is valid for the current IPL or until RVARY ACTIVE is issued. If you just want to deactivate the class(es) without deactivating the RACF database, you should use the command SETROPTS NOCLASSACT (*class-namelist* | *) since INACTIVE NOCLASSACT also deactivates the database. NOCLASSACT is not propagated when issued from a member of a sysplex data sharing group.

INACTIVE (NOTAPE)

specifies that tape volume protection for volumes with IBM standard labels, ANSI labels, and non-standard labels is no longer in effect while RACF is inactive. This option takes effect immediately and is valid for the current IPL or until RVARY ACTIVE is issued. If you just want to deactivate the tape volume without deactivating the RACF database, you should use the command SETROPTS NOCLASSACT(*tapevol*) since INACTIVE (NOTAPE)

also deactivates the database. NOTAPE is not propagated when issued from a member of a sysplex data sharing group.

DATASHARE

specifies to begin data sharing mode. This operand applies only when RACF is enabled for sysplex communication.

If the mode was non-data sharing, RACF discontinues using the hardware RESERVE/RELEASE protocol and uses GRS to serialize access to the RACF database. Systems not already in data sharing mode attempts to connect to all RACF structures. For information on failure scenarios, see *z/OS Security Server RACF System Programmer's Guide*.

If RACF is enabled for sysplex communication, RACF propagates RVARY DATASHARE commands to the other systems in the data sharing group. Members in non-data sharing mode attempt to connect to all structures. If successful, the mode becomes data sharing. Otherwise, it becomes read-only mode.

Note: The current state of the RACF system (that is, ACTIVE or FAILSOFT) from which the command is issued has no effect on either the function or the propagation of the RVARY DATASHARE commands or vice versa, unless the system is in failsoft due to an error during IPL.

NODATASHARE

suspends data sharing mode and enables an installation to update the RACF database even if the system is experiencing coupling facility failure or unavailability. This operand applies only when RACF is enabled for sysplex communication.

When sharing data among many systems, RACF relies on the coupling facility and an alternative serialization technique to provide higher performance. In this environment, RVARY NODATASHARE might result in degraded performance, because RACF discontinues using the coupling facility cache structures and serialization associated with RACF sysplex data sharing and employs the hardware RESERVE/RELEASE protocol. It should be used only to allow critical updates to the database.

If RACF is enabled for sysplex communication, RACF propagates RVARY NODATASHARE commands to the other systems in the data sharing group. The effect of the RVARY DATASHARE command on group members depends on the member's previous database access mode. If the member's previous database access mode was data sharing mode or read-only mode, the member disconnects from all structures and enters non-data sharing mode. If the members previous database access mode was non-data sharing mode, no action is taken.

Note: The current state of the RACF system (that is, ACTIVE or FAILSOFT) from which the command is issued has no effect on either the function or the propagation of the RVARY NODATASHARE commands or vice versa, unless the system is in failsoft due to an error during IPL.

SWITCH

specifies that all processing is to switch from the primary RACF data sets (identified by the DATASET operand) to the corresponding backup data sets. When the switch occurs, the primary data sets are deactivated and deallocated. If you specify DATASET(*) or omit DATASET, the command

RVARY

applies to all primary data sets. If you specify the name of a backup data set on the DATASET operand, RACF issues an error message and ignores the name. In order for the switch to take place, the corresponding backup data sets must be active.

When you issue RVARY SWITCH, RACF associates a set of buffers with the new primary database (the original backup database) and disassociates the buffers from the original primary database (the new backup database). The coupling facility structures associated with the primary and backup RACF databases are not switched, so IRRXCF00_Pnnn structures always correspond to primary database and IRRXCF00_Bnnn structures always correspond to backup database.

To return to the original primary database, you must first activate the backup data sets (the former primary data set) using an RVARY ACTIVE command. An RVARY SWITCH then returns the primary data sets to their original position.

If RACF is enabled for sysplex communication, RACF allocates buffers for backup data sets. The size of the buffer for the backup database is 20 percent of the primary database buffer size. When you issue RVARY SWITCH, RACF associates the larger buffer with the new primary database (the original backup database).

Note: If the data set specified in the SWITCH command is associated with a coupling facility structure failure or with a connection failure, and there is no other failed structure or connection, the system might be put into data sharing mode as a result of the RVARY SWITCH.

DATASET(*dataset-name... | **)

specifies a list of one or more RACF data sets to be switched, reactivated, or deactivated, depending on the SWITCH, ACTIVE, or INACTIVE operands. If you specify DATASET(*) or omit DATASET, the command applies to all primary data sets.

DATASET can be specified with ACTIVE, INACTIVE, or SWITCH; it is not applicable with DATASHARE, NODATASHARE or LIST.

Note: As an exception to normal TSO parsing rules, RACF continues to recognize previously acceptable abbreviations (such as D, DA, DAT, DATA, DATAS) as aliases for DATASET. The shortest acceptable alias for DATASHARE is DATASH.

Do not enclose data set names in single quotation marks.

LIST | NOLIST

LIST

specifies that status information is to be listed for all RACF data sets. If you specify ACTIVE, INACTIVE, SWITCH, DATASHARE, or NODATASHARE, the status displayed is the status after the requested changes have been made if the changes were approved by the operator. If RACF is enabled for sysplex communication, the LIST output includes a line indicating the current operating mode. RVARY LIST does not require operator approval.

The volume information contains an *NA if the device on which the RACF data set resides has been dynamically reconfigured from the system.

NOLIST

specifies that status information for RACF data sets is not to be listed.

Examples

Table 65. RVARY Examples

Example 1	<p><i>Operation</i> User wants to see if the backup data sets are activated.</p> <p><i>Command</i> RVARY LIST</p> <p><i>Output</i> See Figure 53.</p> <p><i>Defaults</i> None</p>
Example 2	<p><i>Operation</i> Operator wants to temporarily deactivate and deallocate RACF to make repairs to a particular primary RACF data set.</p> <p><i>Known</i> The RACF subsystem prefix is #.</p> <p><i>Command</i> #RVARY INACTIVE,DATASET(RACF.PRIM1)</p> <p><i>Output</i> See Figure 54.</p> <p><i>Defaults</i> LIST</p>
Example 3	<p><i>Operation</i> Operator wants to activate the backup data set RACF.BACK1</p> <p><i>Known</i> The backup data set RACF.BACK1 is inactive, and the RACF subsystem prefix is #.</p> <p><i>Command</i> #RVARY ACTIVE,DATASET(RACF.BACK1)</p> <p><i>Output</i> See Figure 55.</p> <p><i>Defaults</i> LIST</p>
Example 4	<p><i>Operation</i> Operator wants to switch from using the primary data set to using the backup data set.</p> <p><i>Known</i> The appropriate backup data set is active, and the RACF subsystem prefix is #.</p> <p><i>Command</i> #RVARY SWITCH,DATASET(RACF.PRIM1)</p> <p><i>Output</i> See Figure 56.</p> <p><i>Defaults</i> LIST</p>
Example 5	<p><i>Operation</i> User wants to change the operating mode to non–data sharing mode for all members of the IRRXCF00 group, in order to allow an update of the RACF data set.</p> <p><i>Known</i> RACF is enabled for sysplex communication but RACF cache structures had not been defined in the coupling facility policy at the time the systems in the group were IPLed. All members of the group are currently in read-only mode.</p> <p><i>Command</i> RVARY NODATASHARE</p> <p><i>Output</i> See Figure 57.</p> <p><i>Defaults</i> LIST</p>
Example 6	<p><i>Operation</i> User wants to change the operating mode from non–data sharing mode to data sharing mode in order to make use of coupling facility performance enhancements.</p> <p><i>Known</i> RACF is enabled for sysplex communication. The user IPLed the system in non–data sharing mode to make use of RVARY and SETROPTS propagation, and is now ready to make use of the coupling facility.</p> <p><i>Command</i> RVARY DATASHARE</p> <p><i>Output</i> See Figure 58.</p> <p><i>Defaults</i> LIST</p>

```

ICH15013I  RACF DATASET STATUS:
           ACTIVE  USE   NUMBER   VOLUME   DATASET
           -----
           YES     PRIM   1       D94RF1   RACF.PRIM1
           NO      BACK   1       D94RF1   RACF.BACK1
           YES     PRIM   2       D94RF1   RACF.PRIM2
           NO      BACK   2       D94RF1   RACF.BACK2
           YES     PRIM   3       D94RF1   RACF.PRIM3
           NO      BACK   3       D94RF1   RACF.BACK3

```

Figure 53. Example 1: Output for the RVARY LIST Command

RVARY

```

ICH15013I  RACF DATASET STATUS:
           ACTIVE  USE   NUMBER  VOLUME   DATASET
           -----
           NO      PRIM   1      *DEALLOC RACF.PRIM1
           NO      BACK   1      D94RF1  RACF.BACK1
           YES     PRIM   2      D94RF1  RACF.PRIM2
           NO      BACK   2      D94RF1  RACF.BACK2
           YES     PRIM   3      D94RF1  RACF.PRIM3
           NO      BACK   3      D94RF1  RACF.BACK3

```

Figure 54. Example 2: Output following Deactivation and Deallocation of RACF.PRIM1

```

ICH15013I  RACF DATASET STATUS:
           ACTIVE  USE   NUMBER  VOLUME   DATASET
           -----
           NO      PRIM   1      *DEALLOC RACF.PRIM1
           YES     BACK   1      D94RF1  RACF.BACK1
           YES     PRIM   2      D94RF1  RACF.PRIM2
           NO      BACK   2      D94RF1  RACF.BACK2
           YES     PRIM   3      D94RF1  RACF.PRIM3
           NO      BACK   3      D94RF1  RACF.BACK3

```

Figure 55. Example 3: Output following the Activation of RACF.BACK1

```

           NO      BACK   1      *DEALLOC RACF.PRIM1
           YES     PRIM   2      D94RF1  RACF.PRIM2
           NO      BACK   2      D94RF1  RACF.BACK2
           YES     PRIM   3      D94RF1  RACF.PRIM3
           NO      BACK   3      D94RF1  RACF.BACK3

```

Figure 56. Example 4: Output following the RVARY SWITCH,DATASET(RACF.PRIM1) Command

```

ICH15019I Initiating propagation of RVARY command to members
           of RACF data sharing group IRRXCF00
ICH15013I RACF DATA SET STATUS:
           ACTIVE  USE   NUMBER  VOLUME   DATASET
           -----
           YES     PRIM   1      D94RF1  RACF.BACK1
           NO      BACK   1      *DEALLOC RACF.PRIM1
           YES     PRIM   2      D94RF1  RACF.PRIM2
           NO      BACK   2      D94RF1  RACF.BACK2
           YES     PRIM   3      D94RF1  RACF.PRIM3
           NO      BACK   3      D94RF1  RACF.BACK3
MEMBER SYS1  IS SYSPLEX COMMUNICATIONS ENABLED &
              IN NON-DATA SHARING MODE.
ICH15020 RVARY command has finished processing.

```

Figure 57. Example 5: Output following the RVARY NODATASHARE Command

```

ICH15019I Initiating propagation of RVARY command to members
          of RACF data sharing group IRRXCF00
ICH15013I RACF DATABASE STATUS:
ACTIVE   USE      NUMBER  VOLUME    DATASET
-----
YES      PRIM      1      D94RF1    RACF.BACK1
NO       BACK      1      *DEALLOC  RACF.PRIM1
YES      PRIM      2      D94RF1    RACF.PRIM2
NO       BACK      2      D94RF1    RACF.BACK2
YES      PRIM      3      D94RF1    RACF.PRIM3
NO       BACK      3      D94RF1    RACF.BACK3
MEMBER SYS1  IS SYSPLEX COMMUNICATIONS ENABLED &
              IN DATA SHARING MODE.
ICH15020 RVARY command has finished processing.

```

Figure 58. Example 6: Output following the RVARY DATASHARE Command

SEARCH (Search RACF Database)

Purpose

Use the SEARCH command to obtain a list of RACF profiles, users, and groups. You can request one or more of the following:

- Profile names that contain a specific character string.
- Profiles for resources that have not been referenced for more than a specific number of days.
- Profiles that RACF recognizes as model profiles.
- Data set and general resource profiles that contain a level equal to or greater than the level you specify.
- User and resource profiles that contain a security label that matches the security label you specify.
- User and resource profiles that contain a security level that matches the security level that you specify.
- User and resource profiles that contain an access category that matches the access category that you specify.
- User profiles that contain an OMVS UID equal to the UID you specify.
- Group profiles that contain an OMVS GID equal to the GID you specify.
- Profiles for tape volumes that contain only data sets with an expiration date that matches the criteria you specify.
- Profiles for data sets that reside on specific volumes (or VSAM data sets that are cataloged in catalogs on specific volumes).
- Profiles for tape data sets, non-VSAM DASD data sets, or VSAM data sets.

You can display the selected profile names at your terminal.

You can also format the selected profile names with specific character strings into a series of commands or messages and retain them in a CLIST data set.

Note: RACF interprets dates with 2 digit years in the following way. YY represents the 2 digit year.

```
IF 70 < YY <= 99 THEN
```

The date is interpreted as 19YY

```
IF 00 <= YY <= 70 THEN
```

The date is interpreted as 20YY

Issuing Options

The following table identifies the eligible options for issuing the SEARCH command:

Table 66. How the SEARCH Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	No	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To obtain information on general resource profiles, see “RLIST (List General Resource Profile)” on page 382.
- To display a data set profile, see “LISTDSD (List Data Set Profile)” on page 200.
- To display a user profile, see “LISTUSER (List User Profile)” on page 223.
- To display a group profile, see “LISTGRP (List Group Profile)” on page 214.

Authorization Required

When issuing the SEARCH command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

You must have a sufficient level of authority for each profile selected as the result of your request, such that one of the following conditions is met:

- You have the SPECIAL attribute,
- You have the system-AUDITOR attribute,
- The profile is within the scope of a group in which you have either the group-SPECIAL or group-AUDITOR attribute, or

If none of the above is true, one of the following must be true:

- If the profile is for a DASD data set, the high-level qualifier of the data set name (or the qualifier supplied by a command installation exit) is your user ID.
- If the profile is in the FILE or DIRECTORY class, the second qualifier of the profile name is your user ID.
- You are on the access list for the profile and you have at least READ authority.
- Your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is on the access list and has at least READ authority.
- You have the OPERATIONS attribute, or the profile is within the scope of a group in which you have the group-OPERATIONS attribute, and the class is DATASET or a general resource class that specifies OPER=YES in the class descriptor table.
- The universal access authority is at least READ (or GLOBAL when listing discrete profiles).

In order to use the USER operand, one of the following must be true:

- You have the SPECIAL or system-AUDITOR attribute.
- You are the owner of the specified user profile.
- You enter your own user ID on the USER operand.
- You have the group-SPECIAL or group-AUDITOR attribute in a group that owns the user profile.

In addition to one of the other four conditions, RACF also checks your security level and categories against those in the specified user profile.

SEARCH

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

Note that it is the authority of the user ID specified on the USER operand that is used to determine if SEARCH displays the profile name.

No authorization is required to the user or group profiles that are listed when the UID or GID keyword is specified.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the command is:

```
[subsystem-prefix]{SEARCH | SR}
    [ AGE(number-of-days) ]
    [ {ALL | GENERIC | NOGENERIC | MODEL | TAPE | VSAM |
      NONVSAM} ]
    [ AT([node].userid ...) | ONLYAT([node].userid ...) ]
    [ {CATEGORY [(category-name)] | EXPIRES(number-of-days) |
      LEVEL(level-number) | SECLABEL [(seclabel-name)] | SECLEVEL
      [(seclabel-name)] | WARNING} ]
    [ CLASS( {DATASET | class-name} ) ]
    [ CLIST ['string-1' ['string-2']] ]
    [ FILTER(filter-string) ]
    [ GID (group-identifier) ]
    [ {LIST | NOLIST} ]
    [ {MASK( {char-1 | *} [ char-2] ) | NOMASK} ]
    [ UID (user-identifier) ]
    [ USER (userid) ]
    [ VOLUME ]
    [ VOLUME(volume-serial) ]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

AGE(*number-of-days*)

specifies the aging factor to be used as part of the search criteria.

Note: This operand works only for discrete profiles and requires that STATISTICS is enabled system-wide.

Only resources that have not been referenced within the specified number of days are selected, unless you specify CLASS(GROUP). In this case, the SEARCH command uses the date on which the group was defined to determine the age.

You can specify up to five digits for *number-of-days*.

ALL | GENERIC | NOGENERIC | MODEL | TAPE | VSAM | NONVSAM

ALL

specifies that RACF is to select all data set profiles (tape, VSAM, and non-VSAM DASD) including both generic and discrete profiles. RACF ignores this operand for classes other than DATASET. ALL is the default if you omit VSAM, NONVSAM, TAPE, GENERIC, NOGENERIC, MODEL, and ALL.

GENERIC | NOGENERIC

specifies whether only generic profiles or no generic profiles (that is, only discrete profiles) are to be selected. If neither operand is specified, both profile types are selected.

RACF ignores these operands unless generic profile command processing is enabled.

MODEL

specifies that only data set profiles having the MODEL attribute are to be selected. RACF ignores this operand for classes other than DATASET.

TAPE

specifies that only tape data sets are to be selected. RACF ignores this operand for classes other than DATASET.

VSAM

specifies that only VSAM data sets are to be selected. RACF ignores this operand for classes other than DATASET.

NONVSAM

specifies that only non-VSAM data sets are to be selected. RACF ignores this operand for classes other than DATASET.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([*node*].*userid* ...)

specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

SEARCH

Note: The SEARCH command is not eligible for command direction when the CLIST keyword is specified. Do not specify the AT and CLIST keywords together on a SEARCH command.

ONLYAT[(*node*).*userid* ...]

SEARCH is not eligible for automatic command direction. If you specify the ONLYAT keyword, the effect is the same as if you specified the AT keyword.

CATEGORY | EXPIRES | LEVEL | SECLEVEL | SECLABEL | WARNING

CATEGORY[(*category-name*)]

specifies that RACF is to select only profiles with an access category matching the category name that you specify, where *category-name* is an installation-defined name that is a member of the CATEGORY profile in the SECDATA class. If you specify CATEGORY and omit *category-name*, RACF selects only profiles that contain undefined access category names (names that were once known to RACF but that are no longer valid).

RACF ignores this operand when CLASS(GROUP) is specified.

EXPIRES(*number-of-days*)

specifies that RACF is to select only tape volumes on which all of the data sets either have expired or will expire within the number of days that you specify. The variable *number-of-days* is a 1-5-digit number in the range of 0 through 65533—or for data sets that never expire, 99999. RACF ignores this operand for classes other than TAPEVOL.

LEVEL(*level-number*)

specifies that RACF is to select only profiles with an installation-defined level that equals the level number you specify. You can specify a value for *level* of 0 through 99.

RACF ignores this operand for classes other than DATASET or classes defined in the RACF class descriptor table.

SECLABEL[(*seclabel-name*)]

specifies that RACF is to select only profiles with a security label name that matches the value you specify for *seclabel*.

SECLEVEL[(*seclabel-name*)]

specifies that RACF is to select only profiles with a security level name that matches *seclabel-name*, where *seclabel-name* is an installation-defined name that is a member of the SECLEVEL profile in the SECDATA class. If you specify SECLEVEL and omit *seclabel-name*, RACF selects only profiles that contain undefined security level names (names that were once known to RACF but that are no longer valid).

RACF ignores this operand when you specify CLASS(GROUP).

WARNING

specifies that only resources with the WARNING indicator are to be selected.

RACF ignores this operand when you specify CLASS as USER or GROUP.

CLASS(**DATASET** | *class-name*)

specifies the name of the class of profiles to be searched. The valid resource classes are DATASET, USER, GROUP, and those specified in the class descriptor table. For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, "Description of RACF Classes" on page 507.

If you omit this operand, the default value is DATASET.

To search all RACF-defined user profiles, you must have either the SPECIAL or AUDITOR attribute.

SEARCH CLASS(USER) can be issued to obtain information about the irrcerta and irrsitec user IDs, which are the user IDs used by RACF to anchor digital certificates.

When searching with the CLASS(GROUP) option, groups are listed based upon the connect authority of the user, **not** READ or higher access to the profile. If CLASS(TAPEVOL) is specified, RACF processes all volumes that meet the search criteria independently, even if the volumes belong to a tape volume set.

If you specify a class that allows profile names to contain any character (ICHERCDE with OTHER=ANY), RACF also selects profiles in the DATASET class that begin with the same first 4 characters as this class name and that satisfy the other criteria you specify on this SEARCH command.

CLIST(['string-1' ['string-2']])

specifies that the selected profile names are to be retained in a CLIST data set. One record is put into the data set for each selected profile name. Profile names containing ampersands (&) appear in the CLIST data set with each occurrence of an ampersand (&) doubled (&&). When the CLIST is executed, double ampersands (&&) prevent the CLIST from performing symbolic substitution when encountering a variable. The CLIST removes only the first ampersand, leaving the second ampersand intact.

'string-1'['string-2']

specifies strings of alphanumeric characters that are put into the CLIST records along with the selected profile names. Each string must be enclosed in single quotes. In this way, you can build a set of commands that are similar except for the profile name.

Mixed case strings are always accepted and preserved for the CLIST operand. If *string-1* is specified, the resulting output CLIST will contain a CONTROL ASIS statement.

The format of the text portion of the CLIST record is as follows:

```
string-1'data-set name'string-2 or
string-1volume-serial-numberstring-2 or
string-1terminal-namestring-2
```

No blank is inserted after *string-1* or before *string-2*. To ensure that the commands execute correctly, use a blank character as the last character in *string-1* and the first character in *string-2*. For example, specify:

```
CLIST('DELDSD ' ' SET')
```

rather than:

```
CLIST('DELDSD' 'SET')
```

An 8-position sequence number is placed on the front of the text.

If both strings are missing, the CLIST record contains only the profile name. If you want a string of data to appear only after the resource name, specify 'string-1' as a double-quotation mark (").

Note: The DASD data set name for the CLIST data set is generated in the format:

```
'prefix.EXEC.RACF.CLIST'
```

SEARCH

where *prefix* is the default data set name prefix in your TSO profile. If you do not have a prefix specified in your TSO profile, (PROFILE NOPREFIX), the userid from the SEARCH command issuer's ACEE is used as the qualifying prefix.

If this data set is partitioned rather than sequential, the CLIST records are placed in member TEMPNAME of the data set. In either case, you can execute the CLIST after SEARCH has finished by issuing the TSO/E command:

```
EXEC 'prefix.EXEC.RACF.CLIST'
```

If a CLIST data set is found through the catalog and is a sequential data set, the records it contains are replaced with the new records. If the CLIST data set is a partitioned data set, however, member TEMPNAME is created to hold the new records, or is replaced if the member already exists.

If the CLIST data set does not already exist, it is created and cataloged. If the CLIST data set created is a partitioned data set, member TEMPNAME is created.

The CLIST data set must have variable length records and a maximum logical record size of 255. This includes a 4-byte length field at the front of the record. The records are numbered in sequence by 10.

Note: The SEARCH command is not eligible for command direction when the CLIST keyword is specified. Do not specify the AT and CLIST keywords together on a SEARCH command.

FILTER(*filter-string*)

(Also see the MASK operand.) specifies the string of alphanumeric characters used to search the RACF database. The filter string defines the range of profile names you wish to select from the RACF database. For a tape or DASD data set name, the filter string length must not exceed 44 characters. For a general resource class, the filter string length must not exceed the length of the profile name specified in the class descriptor table.

Mixed case strings are accepted and preserved when CLASS refers to a class defined in the class descriptor table with CASE=ASIS.

When you issue the SEARCH command with the FILTER operand, RACF lists profile names from the RACF database matching the search criteria specified in the filter string. Note that RACF lists only those profile names that you are authorized to see.

The following generic characters have special meaning when used as part of the filter string:

- % You can use the percent sign to represent any *one* character in the profile name, including a generic character. For example, if you specify DASD%% as a filter string, it can represent profile names such as DASD01, DASD2A, and DASD%5. If you specify %%%% as a filter string, it can represent profile names DASD1, DASD2, DASD%, TAPE%, MY%%%, TAPE*, and %%%%.*
- * You can use a single asterisk to represent *zero or more characters* in a **qualifier**, including generic characters. For example, AB*.CD can represent data set profile names such as AB.CD, ABEF.CD, and ABX.CD. ABC.D* can represent data set profile names such as ABC.DEFG, ABC.D%%%, and ABC.D%*. If you specify a single asterisk as the only character in a

qualifier, it represents the entire qualifier. For example, ABC.* represents data set profile names such as ABC.D, ABC.DEF, ABC.%%%, and ABC.%DE.

- ** For *general resource* and *data set profile names*, you can use a double asterisk to represent zero or more qualifiers in the profile name. For example, AB.**.CD represents data set profile names such as AB.CD, AB.DE.EF.CD, and AB.XYZ.CD. You cannot specify other characters with ** within a qualifier. (For example, you can specify FILTER(USER1.**), but not FILTER(USER1.A**). You can also specify ** as the only characters in the filter-string to represent any entire profile name.

Notes:

1. You can use FILTER for an alternative to MASK | NOMASK as a method for searching the RACF database. FILTER offers more flexibility than MASK. For example, when you use FILTER, you can generalize the character string you specify to match multiple qualifiers or multiple characters within a profile name. You can also specify a character string to match a single character regardless of its value or search for a character string anywhere in a profile name.
2. The SEARCH command might provide unpredictable results when searching on the DIGTCERT or DIGTRING classes. Because these classes contains names with mixed-case characters, the profile filter on the SEARCH command might not function correctly.
3. You cannot use a generic character (*, **, or %) in the high-level qualifier when you define a generic profile for a data set. However, you can use a generic character in the high-level qualifier of a data set name when specifying a filter-string with the FILTER operand.
4. The FILTER and MASK | NOMASK operands are mutually exclusive; you cannot specify FILTER with either MASK or NOMASK on the same SEARCH command.

GID (*group-identifier*)

Specifies that RACF is to display all group profiles which contain the specified *group-identifier* for the GID in the OMVS segment. GID is ignored unless CLASS(GROUP) is specified. When GID is specified, all other keywords (except CLASS) are ignored.

LIST | NOLIST

LIST

specifies that the selected data set names, volume serial numbers, or terminal names are to be displayed at your terminal. LIST is the default value when you omit both LIST and NOLIST.

NOLIST

specifies that the selected data set names, volume serial numbers, or terminal names are not to be displayed at your terminal. You can use this operand only when you specify the CLIST operand. If you use NOLIST without CLIST, the command fails.

MASK | NOMASK

MASK(*char-1* | * [*char-2*])

(Also see the FILTER operand.)

specifies the strings of alphanumeric characters used to search the RACF database. This data defines the range of profile names selected. The two

SEARCH

character strings together must not exceed 44 characters for a tape or DASD data set name, or, for general resource classes, the length specified in the class descriptor table.

char-1

specifies the starting characters of names of profiles to be searched. The string can be any length up to the maximum allowable length of the resource name. All profiles that start with *char-1* in their resource names are selected.

If an asterisk (*) is specified for *char-1*, it specifies that profiles of the search criteria are to be selected:

- For DATASET class, your user ID is used as the mask for the profiles to be selected.
- For other classes, all profiles of the specified class are selected.

char-2

specifies a second string of characters to be included in the search for profiles. All profiles whose names start with *char-1* and contain *char-2* anywhere beyond *char-1* are selected. This limits the list to a subset of the resource names identified with *char-1*.

If asterisk (*) is specified instead of *char-1*, all profiles that contain *char-2* anywhere in their resource names are selected.

If you omit both the MASK and NOMASK operands, this is the same as specifying MASK(*): for the DATASET class, your user ID is used as the mask for profiles to be selected; for other classes, all profiles of the class are selected. (Note also that for classes other than DATASET, omitting both operands is the same as NOMASK).

Mixed case strings are accepted and preserved when CLASS refers to a class defined in the class descriptor table with CASE=ASIS.

NOMASK

specifies that RACF is to select all profiles (to which you are authorized) in the specified class.

Note: The MASK | NOMASK and FILTER operands are mutually exclusive. You cannot specify MASK or NOMASK with FILTER on the same SEARCH command.

UID (*user-identifier*)

Specifies that RACF is to display all user profiles which contain the specified *user-identifier* for the UID in the OMVS segment. UID is ignored unless CLASS(USER) is specified. When UID is specified, all other keywords (except CLASS) are ignored.

USER (*userid*)

specifies that RACF is to list the profiles that the specified user has access to (READ authority or higher, or owner) for the class you specify on the CLASS operand. RACF lists only those profiles that the specified owner is allowed to see.

If you issue:

```
SEARCH USER(JONES) CLASS(ACCTNUM)
```

RACF lists all TSO account numbers that user ID JONES is allowed to use.

If you issue:

```
SEARCH USER(JONES) NOMASK
```

RACF lists profiles in the DATASET class that JONES has access to.

If you issue:

```
SEARCH USER(JONES) CLASS(GROUP)
```

RACF lists all groups that user ID JONES owns or, in which JONES has JOIN or CONNECT authority or the group-SPECIAL attribute.

Notes:

1. If you omit the CLASS operand, the default class is DATASET. For more information, see the description of the CLASS operand.
2. You should not specify a user ID that has been revoked. If you need to display information about a user whose user ID is revoked, perform the following steps:
 - a. Change the password for the user ID
 - b. Resume the user ID
 - c. Issue the SEARCH command to display the desired information
 - d. Revoke the user ID.
3. You can only specify one user ID at a time on the USER operand. If you need to display information about all users, first create a CLIST by issuing the following command:

```
SEARCH CLASS(USER) CLIST('SEARCH USER(' ' ) +
CLASS(class-name)')
```

After you create a CLIST, issue:

```
EXEC 'prefix.EXEC.RACF.CLIST'
```

to display the desired information. (Note that *prefix* is the default data set name prefix in your TSO profile.) For more information, see the description of the CLIST operand.

VOLUME

specifies that you want RACF to display volume information for each tape or DASD data set that meets the search criteria specified by the MASK or FILTER operand.

RACF ignores this operand if you specify GENERIC.

For non-VSAM data sets, the volume serial number displayed is the location of the data set. For VSAM data sets, the volume serial number displayed is the location of the catalog entry for the data set. For tape data sets, the volume serial number displayed is the location of the TVTOC entry for the data set.

This operand is valid only for CLASS(DATASET). RACF ignores it for all other class values.

VOLUME(volume-serial ...)

specifies the volumes to be searched; the volume serial numbers become part of the search criteria. Non-VSAM DASD data sets are selected if they reside on the specified volumes. VSAM data sets are selected if the catalog entries for the data sets reside on the specified volumes. Tape data sets are selected if the TVTOC entries for the data set reside on the specified volumes.

RACF ignores this operand if you specify GENERIC.

SEARCH

If the selected data set names are displayed at your terminal, the volume information is included with each data set name.

This operand is valid only for CLASS(DATASET). RACF ignores it for all other class values.

Examples

Table 67. SEARCH Examples

Example 1	<i>Operation</i>	User CD0 wants to list all of her RACF data set profiles.
	<i>Known</i>	User CD0 is RACF-defined. User CD0 wants to issue the command as a RACF TSO command.
	<i>Command</i>	SEARCH
	<i>Defaults</i>	MASK(CD0) CLASS(DATASET) LIST ALL
	<i>Results</i>	A list of all profiles in the DATASET class beginning with "CD0".
	<i>Example 2</i>	
	<i>Operation</i>	User IA0 wants to remove the RACF profiles for all DATA-type data sets for the group RESEARCH that have not been referenced for 90 days. The user wants a CLIST data set to be created with DELDSD commands for each profile satisfying the search criteria. A list is not desired.
	<i>Known</i>	User IA0 is connected to group RESEARCH (and is the owner of all profiles in group RESEARCH) with the group-SPECIAL attribute. User IA0 wants to issue the command as a RACF TSO command.
	<i>Command</i>	SEARCH FILTER(RESEARCH.DATA) AGE(90) CLIST('DELDSD ') NOLIST
		or
		SEARCH MASK(RESEARCH.DATA) AGE(90) CLIST('DELDSD ') NOLIST
	<i>Defaults</i>	CLASS(DATASET) ALL
	<i>Results</i>	A CLIST data set with the name IA0.EXEC.RACF.CLIST is built, and the records in it are in the format: DELDSD 'data-set-name'
	<i>Example 3</i>	
	<i>Operation</i>	User ADMIN wants to obtain a list of all data set profiles, both discrete and generic, that have the word "DATA" as the second-level qualifier.
	<i>Known</i>	User ADMIN has the SPECIAL attribute. User ADMIN wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.
	<i>Command</i>	@SEARCH FILTER(*.DATA.**)
	<i>Defaults</i>	CLASS(DATASET) LIST ALL
	<i>Results</i>	A list of all profiles in the DATASET class with the word "DATA" as the second-level qualifier. For example, the list might include data sets with names such as RESEARCH.DATA, TEST.DATA, USER.DATA.WEEK1, or GROUP.DATA.TEST.ONE.
	<i>Example 4</i>	
	<i>Operation</i>	User ADM1 wants to obtain a list of all data set profiles, both discrete and generic, having a qualifier (any level) that begins with the word "TEST" and contains only one additional character (such as TEST1, TEST2, or TESTA).
	<i>Known</i>	User ADM1 has the SPECIAL attribute. User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	SEARCH FILTER(**.TEST%.**)
	<i>Defaults</i>	CLASS(DATASET) LIST ALL
	<i>Results</i>	A list of all profiles in the DATASET class having a qualifier of any level that begins with the word "TEST" and contains only one additional character. For example, the list might include data sets with names such as RESEARCH.TEST1, TEST2.DATA, MY.TEST4.DATA, MY.TEST%.* USER.DATA.TEST5, USER.DATA.TEST%.**, or GROUP.DATA.TESTC.FUN.

Table 67. SEARCH Examples (continued)

Example 5	<i>Operation</i>	User ADMIN wishes to find and revoke all user IDs of users who have not accessed the system in the last 90 days. For this to work, the INITSTATS option (specified on the SETROPTS command) must be in effect.
	<i>Known</i>	User ADMIN has the SPECIAL attribute. User ADMIN wants to issue the command as a RACF TSO command.
	<i>Command</i>	SEARCH CLASS(USER) AGE(90) CLIST('ALTUSER ' ' REVOKE')
	<i>Defaults</i>	Process all user ID entries.
	<i>Results</i>	A CLIST data set with the name ADMIN.EXEC.RACF.CLIST listing the user ID for each user that has not accessed the system within 90 days, with records in the following format: ALTUSER userid REVOKE
Example 6	<i>Operation</i>	User ADM1 wants to get a list of all generic profiles for group SALES.
	<i>Known</i>	User ADM1 has the SPECIAL attribute. User ADM1 wants to issue the command as a RACF TSO command.
	<i>Command</i>	SEARCH MASK(SALES.*)
	<i>Defaults</i>	CLASS(DATASET) LIST ALL
	<i>Results</i>	A list of all profiles in the DATASET class beginning with "SALES.*". (Since the string specified contains an asterisk, this list consists only of generic profiles.)
Example 7	<i>Operation</i>	User ADM1 wants to get a list of all data set profiles that include a security level of CONFIDENTIAL. User ADM1 wants to direct the command to run at the local node under the authority of user HICKS.
	<i>Known</i>	User HICKS has the SPECIAL attribute. The CONFIDENTIAL security level has been defined to RACF. User ADM1 wants to issue the command as a RACF TSO command. Users ADM1 and HICKS have an already established user ID association.
	<i>Command</i>	SEARCH CLASS(DATASET) SECLEVEL(CONFIDENTIAL) AT(.HICKS)
	<i>Defaults</i>	LIST ALL Command direction defaults to the local node.
	<i>Results</i>	A list of all profiles in the DATASET class with a security level of CONFIDENTIAL.

SET

Purpose

Use the SET command to:

- List information related to RRSF on the local node
- Specify the name of a member of the RACF parameter library to be processed by RACF
- Set tracing on or off for specified RACF subsystem facilities
- Enable and specify options for automatic direction

Note: You might find it useful to fill out the “Configuration Worksheet” in the *z/OS Security Server RACF System Programmer’s Guide* to help you determine the information you need to issue the SET command.

Issuing Options

The following table identifies the eligible options for issuing the SET command:

Table 68. How the SET Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
No	Yes	No	No	Yes

Note: For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

To define an RRSF node, see “TARGET (Define RRSF Nodes)” on page 481.

Authorization Required

When issuing the SET command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator’s Guide* for further information.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the command is:

subsystem-prefix SET

```

[ AUTOAPPL([
  [ NOTIFY(notify-level(list-of-notify-users)) | NONOTIFY ]
  [ OUTPUT(output-level(list-of-output-users)) | NOOUTPUT ] ) ]
| NOAUTOAPPL ]
[ AUTODIRECT([
  [ NOTIFY(notify-level(list-of-notify-users)) | NONOTIFY ]
  [ OUTPUT(output-level(list-of-output-users)) | NOOUTPUT ] ) ]
| NOAUTODIRECT
]
[ AUTOPWD([
  [ NOTIFY(notify-level(list-of-notify-users)) | NONOTIFY ]
  [ OUTPUT(output-level(list-of-output-users)) | NOOUTPUT ] ) ]
| NOAUTOPWD
]
[ INCLUDE(member-suffix...) ]
[ JESNODE(nodename) ]
[ LIST ]
[ PWSYNC([
  [ NOTIFY(notify-level(list-of-notify-users)) | NONOTIFY ]
  [ OUTPUT(output-level(list-of-output-users)) | NOOUTPUT ] ) ]
| NOPWSYNC
]
[TRACE( {
  [APPC | NOAPPC]
  [ASID(asid ... l*)|NOASID | ALLASIDS]
  [CALLABLE(ALL | NONE | TYPE(type ...))
  | NOCALLABLE]
  [DATABASE({
    [ALL | NONE]
    [ALTER | NOALTER]
    [ALTERI | NOALTERI]
    [READ|NOREAD]])]
  | NODATABASE ]
  [ IMAGE | NOIMAGE ]
  [ JOBNAME(jobname ... l*)
  | NOJOBNAME | ALLJOBNAMES ]
  [PDCALLABLE(ALL | NONE | TYPE(type ...))
  | NOPDCALLABLE]
  [ RACROUTE(ALL | NONE
  | TYPE(type ...) | NORACROUTE ] } ) ]

```

Note: For additional information on issuing this command as a RACF operator command, refer to “Rules for entering RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

SET

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

The settings specified by the SET command only stay in effect while the RACF subsystem address space is active. If the RACF subsystem address space is stopped, no settings are saved. When the RACF subsystem address space is restarted, the default initial values are in effect until overwritten by the new SET commands.

AUTOAPPL | NOAUTOAPPL

AUTOAPPL

specifies that automatic direction of application updates is to be activated. Profiles in the RRSFDATA class control which application updates get automatically directed to which remote nodes. See *z/OS Security Server RACF Security Administrator's Guide* for more information on using the RRSFDATA class to control automatic direction of application updates and for planning information that is necessary before using it.

The operands on the AUTOAPPL keyword specify who gets the result and output from automatically directed application updates.

When specifying node names for OUTPUT and NOTIFY, do not specify nodes running RACF 2.2 if you plan to automatically direct application updates. For descriptions of the OUTPUT and NOTIFY operands and more information about automatic direction, see the AUTODIRECT keyword.

NOAUTOAPPL

specifies that automatic direction of application updates is to be deactivated. This option prevents application update requests from being directed to remote nodes.

The initial value is NOAUTOAPPL.

When SET NOAUTOAPPL is issued, the settings of OUTPUT and NOTIFY are saved. Subsequently, if a SET AUTOAPPL command is issued with no other operands, those settings are restored.

AUTODIRECT | NOAUTODIRECT

AUTODIRECT

specifies that automatic command direction are to be activated. Profiles in the RRSFDATA class control which commands and password synchronization requests get automatically directed to which remote nodes. See *z/OS Security Server RACF Security Administrator's Guide* for more information on using the RRSFDATA class to control automatic direction and for planning information that is necessary before using automatic command direction.

The operands on the AUTODIRECT keyword specify who gets the results and output from automatically directed commands.

When SET AUTODIRECT is issued, the settings of OUTPUT and NOTIFY are saved. Subsequently, if a SET AUTODIRECT command is issued with no other operands, those settings are restored.

If you want to issue the SET AUTODIRECT command to activate both the NOTIFY and OUTPUT settings, both NOTIFY and OUTPUT must be specified on the same command for both to be in effect. If specified on two separate commands, the settings on the first invocation are lost when the second is issued. For example, if you issue SET AUTODIRECT OUTPUT

and then enter SET AUTODIRECT NOTIFY, the OUTPUT setting is lost when the second command is processed. Note that AUTODIRECT NOTIFY and OUTPUT settings are independent of the AUTOAPPL, AUTOPWD, and PWSYNC settings. See the following examples for more information.

When the OUTPUT, NOOUTPUT, NOTIFY, or NONOTIFY keyword is specified, the previous values of *all* of these keywords is overwritten. For example, if the previous setting was:

```
OUTPUT(FAIL(NODEA.ANDREW)) NOTIFY(FAIL(NODEA.ANDREW))
```

and you wanted to also have the command issuer receive FAIL output and command results, you must use:

```
OUTPUT(FAIL(NODEA.ANDREW &RACUID)) NOTIFY(FAIL(NODEA.ANDREW &RACUID))
```

If you just specified:

```
OUTPUT(FAIL(&RACUID))
```

then NODEA.ANDREW would be removed from OUTPUT and NODEA.ANDREW would lose his NOTIFY(FAIL) setting and the value would return to NONOTIFY. Note, however, that these settings for AUTOAPPL, AUTODIRECT, AUTOPWD, and PWSYNC are independent of each other. Continuing the previous example, consider the subsequent settings for SET AUTOPWD:

```
OUTPUT(FAIL(NODEA.ANDREW) NOTIFY(FAIL(NODEA.ANDREW))
```

This resets the table for AUTOPWD, but leaves the previously-specified AUTODIRECT table with its &RACUID intact. AUTODIRECT, AUTOPWD, and PWSYNC are all independent in this way, with regard to the OUTPUT and NOTIFY settings and also about the request's processing.

NOTIFY(*notify-level*(*list-of-notify-users*))

specifies that the user is to be notified (through TSO SEND command) of the results of this RRSF request. The information sent indicates whether the command was successful or unsuccessful, but does not include other details about the request's processing.

ALWAYS

specifies that results or output from all requests of this RRSF function are to be returned to specified users. This option should be used if the users are interested in the results of *every* request. Output includes informational, warning, and error messages.

WARN

specifies that, in the case of AUTODIRECT, results or output from automatically directed commands are to be returned to the specified users only when the return code from the command is 4 or greater. In the case of AUTOAPPL, AUTOPWD, and PWSYNC, WARN is equivalent to FAIL.

FAIL

specifies that, in the case of AUTODIRECT, results or output from automatically directed commands are to be returned to the specified users only when the return code from the command is 8 or greater. For AUTOAPPL, AUTOPWD, and PWSYNC, results or output from the request are to be returned to the specified users whenever the return code from the request is non-zero.

list-of-notify-users is up to four users who are to receive output and/or notification of results. A user can be specified in one of the following ways:

- | | |
|--------------------|--|
| node.userid | The node and user ID separated by a period. For AUTOAPPL, do not specify nodes running RACF 2.2. |
| .userid | The name of the user ID on the local node preceded by a period |
| &RACUID | <p>Original issuer with regard to node and user</p> <p>When &RACUID is specified, where the results and output are sent depend on the situation. Consider the following scenarios:</p> <ul style="list-style-type: none"> • For password synchronization, automatic password direction, and automatic direction of application updates, results and output go to the specific system. For automatic command direction, the results and output go to the MAIN system. For a multi-system node, the MAIN system might not necessarily be the specific system of that issuing node. • A user on node A directs a command to node B which results in automatic command direction to node C. &RACUID specified on node C for AUTODIRECT NOTIFY or OUTPUT sends data to node A. • A user on node A directs a command to node B which results in a password change. This password change is propagated by password synchronization to node C. &RACUID specified on node C for PWSYNC NOTIFY or OUTPUT sends data to node B. • A user on node A changes a password, such that automatic password direction updates the corresponding userid on node B. This userid propagates the password change to a peer association on that same node B. &RACUID specified on node B for PWSYNC NOTIFY or OUTPUT would send data to the original userid on node B (not A). |

Note: If &RACUID is specified along with the user ID from which you are issuing the command, password change (covered by password synchronization or automatic password direction), or application update, that user ID receives the output or notification twice.

If you plan to use &RACUID for application updates, be aware of the following:

- IBM recommends that installations not use &RACUID for AUTOAPPL output or notification. Application updates can be done under undefined users, revoked user IDs, or the user ID of the RACF address space, which can produce unexpected results during output delivery, including lost output and error messages on the console.

- Ensure all possible user ID destinations through &RACUID have the authority to create data sets. For example, an installation would not want to use &RACUID for AUTOPWD or PWSYNC if the original issuer of a password change could be a CICS user, who is unlikely to have authority to create the RRSFLIST output data set.
- If the ACEE= keyword is used on the macro, the output goes to the userid associated with the ACEE= keyword, not to the user ID of the task or address space that actually issued the macro.

The SET command does not perform existence checking for either the user ID or node.

The combination of users specified in the list of notify users variables can be up to a maximum of four different users. In other words, the cumulative total of unique users cannot exceed 4 in both the OUTPUT and NOTIFY keywords. The same four users can be specified in each list; however, if four users are specified on one of the keywords, a fifth user cannot be specified on the other keyword. For example, if four users are specified on the OUTPUT keyword, a fifth user cannot be specified on the NOTIFY keyword.

NONOTIFY

specifies that no TSO SEND commands are issued with the results of the RRSF request.

The initial value is NONOTIFY.

The allowed values for *notify-level* are:

OUTPUT(*output-level*(*list-of-output-users*))

specifies that the output from the RRSF request should be put in the RRSFLIST data set for the user named on this keyword. If the output cannot be put in the RRSFLIST data set for any reason, the output is transmitted to the user.

Because LIST-type commands are ineligible for automatic command direction, the output usually contains messages issued during command processing, such as informational, warning, or error messages.

The valid values for *output-level* are the same as those described for *notify-level* with the NOTIFY keyword.

The valid values for *list-of-output-users* are the same as those described for *list-of-notify-users* with the NOTIFY keyword.

NOOUTPUT

specifies that no output, warning, or error messages are kept or sent to anyone.

The initial value is NOOUTPUT.

NOAUTODIRECT

specifies that automatic command direction is to be deactivated.

This option prevents commands from being automatically directed to remote nodes.

The initial value is NOAUTODIRECT.

SET

When SET NOAUTODIRECT is issued, the settings of OUTPUT and NOTIFY are saved. Subsequently, if a SET AUTODIRECT command is issued with no other operands, those settings are restored.

AUTOPWD | NOAUTOPWD

AUTOPWD

specifies that automatic password direction is to be activated. Profiles in the RRSFDATA class, "AUTODIRECT.nodename.USER.PWSYNC", control which automatic password directed requests get directed to which remote nodes. See the *z/OS Security Server RACF Security Administrator's Guide* for more information on using the RRSFDATA class to control automatic password direction and for planning information that is necessary before using automatic password direction.

The operands on the AUTOPWD keyword specify who gets the result and output from automatically directed passwords. Refer to the descriptions of OUTPUT and NOTIFY under the AUTODIRECT keyword.

NOAUTOPWD

specifies that automatic password direction is to be deactivated.

This option prevents passwords from being automatically directed to remote nodes.

The initial value is NOAUTOPWD.

When SET NOAUTOPWD is issued, the settings of OUTPUT and NOTIFY are saved. Subsequently, if a SET AUTOPWD command is issued with no other operands, those settings are restored.

INCLUDE(member-suffix...)

provides the ability to specify that the contents of one or more members of the RACF parameter library are to be processed. The INCLUDE keyword provides a convenient mechanism to process a previously defined set of RACF commands, such as SET and TARGET.

One or more suffixes can be specified with the INCLUDE keyword. Each specified suffix must be:

- 1 or 2 characters in length
- In the alphanumeric character set (A-Z, 0-9, # (X'7B') , \$ (X'5B') or @ (X'7C'))

such that when the suffix is appended to IRROPT it results in the name of a member of the RACF parameter library.

SET INCLUDE commands can be nested in members of the RACF parameter library. For example, if member IRROPT01 contains a SET INCLUDE IRROPT02 command, then member IRROPT02 can contain a SET INCLUDE IRROPT03 command.

Be careful to adhere to a hierarchical order when nesting SET INCLUDE commands. For example, if IRROPT01 contains a SET INCLUDE IRROPT02 command, then IRROPT01 cannot contain a SET INCLUDE IRROPT01 command. Also, if IRROPT02 contains a SET INCLUDE command for any other parameter library members, those members cannot contain a SET INCLUDE IRROPT01 command. This restriction exists to prevent a never-ending loop of inclusion.

If a suffix appended to IRROPT does not result in the name of a member of the RACF parameter library, a message is issued and that suffix is ignored.

If the INCLUDE keyword is specified with any other SET keywords, the included members are processed first. The values specified for the other keywords override any values specified for the keywords in the included members. For example, the values specified for TRACE override any trace values specified in the included members.

No authorization checking or auditing is done for the commands in included members.

JESNODE(nodename)

specifies the name of the node needed by RRSF in the cases where returned output from the directed commands must be transmitted to the user. RRSF queries the primary JES system during initialization in an attempt to obtain this name automatically. This keyword should be used in the cases where RRSF cannot automatically obtain this name.

No validity checking is done on the value specified with the JESNODE keyword.

LIST

lists the attributes of an RRSF node and trace options.

The LIST keyword provides the ability to obtain information about the RRSF node's configuration, status related to the RACF subsystem, and status of the trace options set for ASID, CALLABLE, DATABASE, JOBNAME, PDCALLABLE, and RACROUTE trace keywords.

The LIST keyword can be specified alone or in combination with other SET command keywords. When used in combination with other SET command keywords, the information displayed reflects the results after processing the other keywords.

Note: LIST is the default, if the SET command is issued with no keywords.

PWSYNC | NOPWSYNC

PWSYNC

specifies that password synchronization is to be activated. See the *z/OS Security Server RACF Security Administrator's Guide* for more information on using the RRSFDATA class to control password synchronization and for planning information that is necessary before using password synchronization. For information on how to establish password synchronization between user IDs review the RACLINK section of this manual.

The operands on the PWSYNC keyword specify who gets the result and output from password changes covered by password synchronization. Refer to the descriptions of OUTPUT and NOTIFY under the AUTODIRECT keyword.

NOPWSYNC

specifies that synchronized password processing is to be deactivated.

The initial value is NOPWSYNC.

When SET NOPWSYNC is issued, the settings of OUTPUT and NOTIFY are saved. Subsequently, if a SET PWSYNC command is issued with no other operands, those settings are restored.

TRACE

specifies whether tracing is to take place for the following events, using the generalized trace facility (GTF).

SET

If the TRACE keyword is specified, at least one subkeyword must be specified to indicate whether or not tracing is to be turned on or off for each of these events. There are no defaults for these event types. For example, if APPC is the only operand specified, then the current setting for tracing IMAGE events is not changed. If IMAGE tracing was in effect, it remains in effect. Likewise, if NOIMAGE had been in effect, it would remain in effect.

The initial value is NOAPPC, NOASID, NOCALLABLE, NODATABASE, NOIMAGE, NOJOBNAME, and NORACROUTE.

The SET LIST command should always be used to verify the trace parameters have been set as expected.

The trace records are intended for use in consultation with the IBM support center when diagnosing potential RACF subsystem problems. For more information, see *z/OS Security Server RACF Diagnosis Guide*.

Attention: Trace records might contain passwords and therefore trace output data sets should be appropriately protected.

APPC | NOAPPC

APPC

tracing is to be in effect for APPC events. The trace information contains the APPC transaction return code and reason code.

NOAPPC

tracing is not to be in effect for APPC events.

ASID(*asid* ...) | NOASID | ALLASIDS

ASID(*asid* ...)

use ASID to establish that you wish to trace the included list of address spaces for the CALLABLE, DATABASE, and RACROUTE keywords. When ASID(*asid* ...) is specified, by consecutive invocations of the SET command, the *asid* list is deleted and rebuilt with the current specification each time the command is issued.

NOASID

use NOASID to eliminate the list of addresses to enable tracing all address spaces.

ALLASIDS

tracing is in effect for all address spaces.

DATABASE | NODATABASE

DATABASE

use to trace RACF database manager requests.

ALL | NONE

use ALL to enable tracing of all requests.

use NONE to disable tracing.

ALTER | NOALTER

use ALTER to enable tracing all RACF database manager calls that change the database. Calls included are RENAMEs, ALTERs, ADDs and DELETEs. No further granularity is provided for calls that change the database.

use NOALTER to disable tracing of all RENAMEs, ALTERs, ADDs and DELETEs.

ALTERI | NOALTERI

use ALTERI to enable tracing all RACF database manager calls that change fields in the database that use ALTERI as the request.

use NOALTERI prevent the tracing of these requests.

READ | NOREAD

use READ to enable tracing all RACF database manager RACF calls that locate profiles in the database.

use NOREAD prevent the tracing of these requests.

NODATABASE

use NODATABASE to disable tracing database manager requests; equivalent to DATABASE(NONE).

CALLABLE | NOCALLABLE**CALLABLE**

use to trace z/OS UNIX System Services calls.

ALL | NONE | TYPE(*type ...*)

use to control the degree of tracing.

ALL

use to enable tracing of all z/OS UNIX calls.

NONE

use to reset tracing.

TYPE(*type ...*)

use to enable tracing of one or more specific z/OS UNIX calls. The request types that are supported are listed in the following table:

Callable service	Service / Type Number	Callable service	Service / Type Number
IRRSIU00	1	IRRSMR00	25
IRRSDU00	2	IRRSPT00	26
IRRSMF00	3	IRRSUG00	27
reserved	4	IRRSFK00	28
IRRSMM00	5	IRRSM100	29
IRRSKA00	6	IRRSK100	30
IRRSKP00	7	IRRSK100	31
IRRSUM00	8	IRRSK200	32
IRRSKM00	9	IRRSK300	33
IRRSKG00	10	IRRSK400	34
IRRSSU00	11	IRRSK500	35
IRRSKU00	12	IRRSK600	36
IRRSSG00	13	IRRSK700	37
IRRSKG00	14	IRRSK800	38
IRRSKO00	15	IRRSK900 *	39
IRRSKF00	16	IRRSK1000	40
IRRSKA00	17	IRRSK1100	41
IRRSKX00	18	IRRSK1200	42

SET

Callable service	Service / Type Number	Callable service	Service / Type Number
IRRSAU00	19	IRRSPK00	43
IRRSKO00	20	IRRSPX00	44
IRRSQS00	21	IRRSCH00	45
IRRSQF00	22	IRRSPY00	46
IRRSXS00	23	IRRSCL00	47
IRRSKF00	24		

* Callable Service IRRSEQ00, R_Admin, has its own trace facility. For more information, see *z/OS Security Server RACF Diagnosis Guide*.

The TYPE operand is cumulative; issuing NOCALLABLE or CALLABLE(NONE) will reset the trace.

NOCALLABLE

use NOCALLABLE to reset the trace; equivalent to CALLABLE(NONE).

IMAGE | NOIMAGE

IMAGE

tracing is to be in effect for IMAGE events. The trace information contains the command image being processed.

NOIMAGE

tracing is not to be in effect for IMAGE events.

JOBNAME | ALLJOBNAMES | NOJOBNAME

JOBNAME(jobname ...)

use JOBNAME to establish that you wish to trace the included list of jobs for CALLABLE, DATABASE, and RACROUTE traces. The *jobname* variable is a list of one or more jobs to be traced. The *jobname* variable will accept * at the end of *jobname* so as to allow matching on a set of jobnames. For example, if z/OS UNIX created jobnames of MAPES, MAPES2 and MAPES3, tracing would be facilitated through the use of * at the end of the jobname, as in JOBNAME(MAPES*).

When JOBNAME(*jobname* ...) is specified, by consecutive invocations of the SET command, the *jobname* list is deleted and rebuilt with the current specification each time the command is issued.

ALLJOBNAMES

tracing is to be in effect for all job names.

NOJOBNAME

use to resets tracing.

PDCALLABLE | NOPDCALLABLE

PDCALLABLE

use to trace IBM Policy Director Authorization Services SAF calls.

ALL | NONE | TYPE(type ...)

use to control the degree of tracing.

ALL

use to enable tracing of all IBM Policy Director Authorization Services SAF calls.

NONE

use to reset tracing.

TYPE(type ...)

use to enable tracing of one or more specific IBM Policy Director Authorization Services SAF calls. The request types that are supported are listed in the following table:

CALLABLE SERVICE	Service / Type Number
IRRSZA00	1
IRRSZC00	2

The TYPE operand is cumulative; issuing NOPDCALLABLE or PDCALLABLE(NONE) will reset the trace.

NOPDCALLABLE

use NOPDCALLABLE to reset the trace; equivalent to PDCALLABLE(NONE).

RACROUTE | NORACROUTE**RACROUTE**

use to trace RACROUTE calls.

ALL | NONE | TYPE(type ...)

use to control the degree of tracing.

ALL

use to enable tracing of all RACROUTE calls.

NONE

use to reset tracing.

TYPE(type ...)

use to enable tracing of one or more specific RACROUTE calls. The request types that are supported are listed in the following table:

RACROUTE REQUEST=	Service / Type Number
AUTH	1
FASTAUTH	2
LIST	3
DEFINE	4
VERIFY	5
EXTRACT	6
DIRAUTH	7
TOKENMAP	8
VERIFYX	9
TOKENXTR	10
TOKENBLD	11
EXTRACT, BR=YES	12

SET

RACROUTE REQUEST=	Service / Type Number
AUDIT	13
STAT	14
SIGNON	15
TOKENMAP, XMEM	16
TOKENXTR, XMEM	17

The TYPE operand is cumulative; issuing NORACROUTE or RACROUTE(NONE) will reset the trace.

NORACROUTE

use NORACROUTE to reset the trace; equivalent to RACROUTE(NONE).

Examples

Table 69. SET Examples

Example 1

Operation User ADMIN wants to enable automatic command direction and establish that LAURIE at POKMVS and the command issuer receives output and notification when an automatically directed command receives a return code of 8 or greater.

Known The RACF subsystem prefix is @.

Command @SET AUTODIRECT (OUTPUT (FAIL (POKMVS.LAURIE &RACUID))
NOTIFY (FAIL (POKMVS.LAURIE &RACUID)))

Defaults None.

Example 2

Operation User ADMIN wants to enable automatic command direction and establish that:

- ACDERROR at POKMVS will receive warning and error output, but no TSO SEND messages.
- ANDREW at POKMVS will receive warning output, error output, and TSO SEND messages for error conditions.
- LAURIE at POKMVS will not receive any output, but will receive TSO SEND messages for error conditions.
- The command issuer gets no notification of automatically directed commands.

Known The RACF subsystem prefix is @. LAURIE at POKMVS has the ability to browse the RRSFLIST data set of ACDERROR at POKMVS to determine what needs to be fixed.

Command @SET AUTODIRECT (OUTPUT (WARN (POKMVS.ACDERROR POKMVS.ANDREW))
NOTIFY (FAIL (POKMVS.LAURIE POKMVS.ANDREW)))

Defaults None

Example 3

Operation User ADMIN wants to enable automatic direction, automatic password direction, automatic direction of application updates, but not password synchronization. User ADMIN wants to be notified and receive output for all failures. The command issuer needs to always receive notification and output for automatically directed commands (but not for automatic password direction or automatic direction of application updates).

Known The RACF subsystem prefix is @.

Commands @SET AUTODIRECT (OUTPUT (FAIL (POKMVS.ADMIN &RACUID))
NOTIFY (FAIL (POKMVS.ADMIN &RACUID)))
@SET AUTOPWD (OUTPUT (FAIL (POKMVS.ADMIN))
NOTIFY (FAIL (POKMVS.ADMIN)))
@SET AUTOAPPL (OUTPUT (FAIL (POKMVS.ADMIN))
NOTIFY (FAIL (POKMVS.ADMIN)))

Defaults None

Table 69. SET Examples (continued)

Example 4	<i>Operation</i>	User ADMIN wants to enable tracing for all verifys issued in a particular address space.
	<i>Known</i>	The RACF subsystem prefix is @.
	<i>Command</i>	@SET TRACE(RACROUTE(TYPE(2,5,9)) ASID(17))
	<i>Defaults</i>	None
Example 5	<i>Output</i>	See Figure 59
	<i>Operation</i>	User ADMIN wants to obtain information concerning the RRSF node's configuration and status related to the RACF subsystem. User ADMIN also wants to turn on tracing for IMAGE events.
	<i>Known</i>	Since the LIST keyword is used in combination with the TRACE keyword, the information displayed reflects the results after processing the TRACE keyword.
		The RACF subsystem prefix is @.
Example 6	<i>Command</i>	@SET LIST TRACE(IMAGE)
	<i>Defaults</i>	None
	<i>Output</i>	See Figure 59.
	<i>Operation</i>	User ADMIN wants to enable tracing for the aznAccess SAF callable service for jobname IBMUSER.
Example 7	<i>Known</i>	The RACF subsystem prefix is @.
	<i>Command</i>	@SET TRACE(PDCALLABLE(TYPE(1)) JOBNAME(IBMUSER))
	<i>Defaults</i>	None
	<i>Output</i>	See Figure 60.
Example 7	<i>Operation</i>	User ADMIN wants to verify that tracing has been enabled on this RRSF node for the aznAccess z/OS Policy Director SAF callable service
	<i>Known</i>	The RACF subsystem prefix is @.
	<i>Command</i>	@SET LIST
	<i>Output</i>	SET LIST output indicates that tracing has been enabled for PDCALLABLE service request type 1, aznAccess. See Figure 61.

```

RACFR12 IRRH005I (@) RACF SUBSYSTEM INFORMATION:
TRACE OPTIONS
- IMAGE
- NOAPPC
- RACROUTE
  2 5 9
- NOCALLABLE
- NODATABASE
- ASID
  17
- NOJOBNAME
SUBSYSTEM USERID - IBMUSER
JESNODE (FOR TRANSMITS) - POKVMMCL
AUTOMATIC COMMAND DIRECTION IS *NOT* ALLOWED
AUTOMATIC PASSWORD DIRECTION IS *NOT* ALLOWED
PASSWORD SYNCHRONIZATION IS *NOT* ALLOWED
AUTOMATIC DIRECTION OF APPLICATION UPDATES IS *NOT* ALLOWED
RACF STATUS INFORMATION:
  TEMPLATE VERSION - HRF7705
  DYNAMIC PARSE VERSION - HRF7705

```

Figure 59. Example 3: Output for the SET LIST Command

```

IRRH004I (@) RACF SUBSYSTEM SET COMMAND HAS COMPLETED SUCCESSFULLY.

```

Figure 60. EXAMPLE 6: Response from the SET TRACE command for PDCALLABLE service.

SET

```
IRRH005I (@) RACF SUBSYSTEM INFORMATION:  
TRACE OPTIONS    -NOIMAGE  
                -APPC  
                -PDCALLABLE  
                  1  
...
```

Figure 61. EXAMPLE 7: Output for the SET LIST Command for PDCALLABLE service.

SETROPTS (Set RACF Options)

Purpose

Use the SETROPTS command to set system-wide RACF options related to resource protection dynamically. Specifically, you can use SETROPTS to do the following:

- Gather and display RACF statistics
- Protect terminals
- Log RACF events
- Permit list-of-groups access checking
- Display options currently in effect
- Enable or disable the generic profile checking facility on a class-by-class basis or for all classes system-wide
- Control user password expiration interval
- Establish password syntax rules
- Activate password processing for checking previous passwords and limiting incorrect password attempts
- Activate auditing for access attempts by class
- Activate auditing for security labels
- Require that all work entering the system, including users logging on and batch jobs, have a security label assigned
- Enable or disable the global access checking facility
- Refresh in-storage profile lists and global access checking tables
- Set the password the operator must supply in order for RACF to complete an RVMRY command that changes RACF status or changes the RACF databases
- Enable or disable the sharing, in common storage, of discrete and generic profiles for general resource classes
- Activate or deactivate auditing of access attempts to RACF-protected resources based on installation-defined security levels
- Control the automatic data set protection (ADSP) attribute for users
- Activate profile modeling for GDG, group, and user data sets
- Activate protection for data sets with single-level names
- Control logging of real data set names
- Control the job entry subsystem options
- Activate tape data set protection
- Control whether RACF is to allow users to create or access data sets that do not have RACF protection
- Activate and control the scope of erase-on-scratch processing
- Activate program control, which includes both access control to load modules and program access to data
- Prevent users from accessing uncataloged permanent data sets
- Establish a system-wide VTAM session interval
- Set an installation-wide default for the RACF security retention period for tape data sets
- Activate enhanced generic naming for data sets and entries in the global access checking table
- Set installation defaults for primary and secondary national languages.

SETROPTS

- Activate auditing for APPC transactions

If you specify the AUDIT operand, RACF logs all uses of the RACROUTE REQUEST=DEFINE SVC and all changes made to profiles by RACF commands. Following are the classes that can be specified in the AUDIT operand and the commands and SVCs that is logged for each class:

USER	GROUP	DATASET	Class Descriptor Table Entries
ADDUSER	ADDGROUP	ADDSD	PERMIT
ALTUSER	ALTGROUP	ALTDSD	REQUEST=DEFINE SVC
CONNECT	CONNECT	DELDSD	RALTER
DELUSER	DELGROUP	PERMIT	RDEFINE
PASSWORD	REMOVE	REQUEST=DEFINE SVC	RDELETE
REMOVE			

Most RACF functions do not require special versions or releases of the operating system or operating system components. Some, however, do require that your system be at a certain level.

Using SETROPTS when RACF is enabled for sysplex communication: When RACF is enabled for sysplex communication, RACF propagates the following SETROPTS commands:

- GENERIC REFRESH
- GLOBAL
- GLOBAL REFRESH
- RACLIST
- NORACLIST
- RACLIST REFRESH
- WHEN(PROGRAM)
- WHEN(PROGRAM) REFRESH

When issued from a member of the RACF data sharing group, these commands, if successful on the member that issues them, are propagated in a controlled, synchronized manner to the other members in the group. A system in read-only mode can participate if it receives a SETROPTS command propagated from another system, but a user on a system in read-only mode cannot issue any SETROPTS commands except for the SETROPTS LIST command. For propagated SETROPTS REFRESH commands, members of the data sharing group are notified to either create, update, or delete some in-storage information. These commands are coordinated to ensure that all systems begin to use the changed information simultaneously, and to always see a consistent view of this information.

RACF serializes propagated SETROPTS commands to prevent conflicting commands of the same type (for example, SETROPTS RACLIST and SETROPTS NORACLIST) from processing simultaneously.

Refer to the specific parameter descriptions for additional information about using these parameters.

Notes:

1. The options you specify on SETROPTS are common on systems that share the RACF database. All the systems involved must have the required levels of software. If you activate the SECLABEL and ML options on one system, they are activated on all systems.
2. If RACF is not enabled for sysplex communication, the SETROPTS commands that would be propagated to all members of a data sharing group must instead be issued on each system sharing the database. Although the command is not propagated, RACF does record the fact that a SETROPTS RACLIST was issued. The next time that any system sharing the database is IPLed, the SETROPTS RACLIST is done on that sharing system.
3. When the SETROPTS command is from ISPF, the TSO command buffer (including password data) is written to the ISPLOG data set. As a result, you should not issue the SETROPTS command from ISPF or you must control the ISPLOG data set carefully.
4. If the SETROPTS command is issued as a RACF operator command, the command and the password data is written to the system log. Therefore, use of SETROPTS as a RACF operator command should either be controlled or you should issue the command as a TSO command.

Note: RACF interprets dates with 2 digit years in the following way. YY represents the 2 digit year.

```
IF 70 < YY <= 99 THEN
```

The date is interpreted as 19YY

```
IF 00 <= YY <= 70 THEN
```

The date is interpreted as 20YY

Issuing Options

The following table identifies the eligible options for issuing the SETROPTS command:

Table 70. How the SETROPTS Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
Yes	Yes	Yes	Yes	Yes

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Authorization Required

When issuing the SETROPTS command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

SETROPTS

Most SETROPTS command functions require you to have the SPECIAL or AUDITOR attributes.

If you have the SPECIAL attribute you can use all of the operands except those listed below which require the AUDITOR attribute:

- APPLAUDIT | NOAPPLAUDIT
- AUDIT | NOAUDIT
- CMDVIOL | NOCMDVIOL
- LOGOPTIONS
- OPERAUDIT | NOOPERAUDIT
- SAUDIT | NOSAUDIT
- SECLABELAUDIT | NOSECLABELAUDIT
- SECLEVELAUDIT | NOSECLEVELAUDIT

If you have either the SPECIAL or AUDITOR attribute, you can use the LIST operand.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified *node.userid* pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the *userid* specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified *node.userid* pair(s) if the user IDs are not identical.

In some situations, you can use SETROPTS even if you do not have the SPECIAL or AUDITOR attributes. These situations are:

- You can specify the LIST operand if you have the group-SPECIAL or group-AUDITOR attribute in the current connect group or if GRPLIST is active in any group that you are connected to.
- You can specify REFRESH together with GENERIC if you have the group-SPECIAL, AUDITOR, group-AUDITOR, OPERATIONS, or group-OPERATIONS attribute, or CLAUTH authority for the classes specified.
- You can specify REFRESH together with GLOBAL if you have the OPERATIONS attribute or CLAUTH authority for the classes specified.
- You can specify REFRESH together with RACLIST if you have CLAUTH authority to the specified class. You can specify REFRESH together with WHEN(PROGRAM) if you have CLAUTH authority for the program class.

Note: The syntax diagram does not indicate the defaults that are in effect when RACF is using a newly-initialized database. You can find these defaults in the description of each operand. As you establish the system-wide defaults your installation needs, you might find it useful to mark the syntax diagram to reflect your choices.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the command is:

<i>[subsystem-prefix]</i> {SETROPTS SETR}

```

[ ADDCREATOR | NOADDCREATOR ]
[ ADSP | NOADSP ]
[ APPLAUDIT | NOAPPLAUDIT ]
[ AT([node].userid ...) | ONLYAT([node].userid ...) ]
[ {AUDIT | NOAUDIT} ({class-name ... | *}) ]
[ CATDSNS ( FAILURES | WARNING ) | NOCATDSNS ]
[ {CLASSACT | NOCLASSACT} ( {class-name... | *} ) ]
[ CMDVIOL | NOCMDVIOL ]
[ COMPATMODE | NOCOMPATMODE ]
[ EGN | NOEGN ]
[ EIMREGISTRY | NOEIMREGISTRY ]
[ ERASE [(
    { ALL
    | SECLEVEL(seclevel-name)
    | NOSECLEVEL}
)]
| NOERASE ]
[ {GENCMD | NOGENCMD} ( {class-name... | *} ) ]
[ {GENERIC | NOGENERIC} ( {class-name... | *} ) ]
[ GENERICOWNER | NOGENERICOWNER ]
[ {GENLIST | NOGENLIST} (class-name ...) ]
[ {GLOBAL | NOGLOBAL} ( {class-name ... | *} ) ]
[ GRPLIST | NOGRPLIST ]
[ INACTIVE(unused-userid-interval) | NOINACTIVE ]
[ INITSTATS | NOINITSTATS ]
[ JES (
    [ BATCHALLRACF | NOBATCHALLRACF ]
    [ EARLYVERIFY | NOEARLYVERIFY ]
    [ XBMALLRACF | NOXBMALLRACF ]
    [ NJEUSERID(userid) ]
    [ UNDEFINEDUSER(userid) ]
) ]
[ KERBLVL(0|1)]
[ LANGUAGE(
    [ PRIMARY(language) ]
    [ SECONDARY(language) ]
) ]
[ LIST ]
[ LOGOPTIONS(
    {ALWAYS(class-name, ...), ...
    | NEVER(class-name, ...), ...
    | SUCCESSES(class-name, ...), ...
    | FAILURES(class-name, ...), ...
    | DEFAULT( {class-name, ... | *} ) }
) ]
[ MLACTIVE [( FAILURES | WARNING )] | NOMLACTIVE ]
[ MLQUIET | NOMLQUIET ]
[ MLS [( FAILURES | WARNING )] | NOMLS ]
[ MLSTABLE | NOMLSTABLE ]
[ MODEL(
    [ GDG | NOGDG ]
    [ GROUP | NOGROUP ]
    [ USER | NOUSER ]
) | NOMODEL ]
[ OPERAUDIT | NOOPERAUDIT ]

```

SETROPTS

```
[ PASSWORD(
  [ HISTORY(number-previous-passwords)
    | NOHISTORY ]
  [ INTERVAL(password-change-interval) ]
  [ REVOKE(number-invalid-passwords)
    | NOREVOKE ]
  [ {RULEn(LENGTH(m1:m2) content-keyword (position))
    | NORULEn
    | NORULES} ]
  [ WARNING(days-before-password-expires)
    | NOWARNING ]
) ]
[ PREFIX(prefix) | NOPREFIX ]
[ PROTECTALL [( FAILURES | WARNING )] | NOPROTECTALL ]
[ {RACLIST | NORACLIST} (class-name ...) ]
[ REALDSN | NOREALDSN ]
[ REFRESH ]
[ RETPD(nnnnn) ]
[ RVARYPW( [SWITCH(switch-pw)] [STATUS(status-pw)] ) ]
[ SAUDIT | NOSAUDIT ]
[ SECLABELAUDIT | NOSECLABELAUDIT ]
[ SECLABELCONTROL | NOSECLABELCONTROL ]
[ SECLEVELAUDIT (security-level) | NOSECLEVELAUDIT ]
[ SESSIONINTERVAL(n) | NOSESSIONINTERVAL ]
[ {STATISTICS | NOSTATISTICS} ({class-name... | *}) ]
[ TAPEDSN | NOTAPEDSN ]
[ TERMINAL( NONE | READ ) ]
[ {WHEN | NOWHEN} (PROGRAM) ]
```

Note: For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the execution environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

ADDCREATOR | NOADDCREATOR

ADDCREATOR

specifies that if a user defines any new DATASET or general resource profile using ADDSD, RDEFINE or RACROUTE REQUEST=DEFINE, the profile creator's user ID is placed on the profile access list with ALTER authority.

NOADDCREATOR

specifies that if a user defines any new DATASET or general resource profile using ADDSD, RDEFINE or RACROUTE REQUEST=DEFINE, or creates discrete profiles other than DATASET and TAPEVOL using RACROUTE REQUEST=DEFINE, RACF does not place the profile creator's user ID on the profile's access list. If the profile creator uses profile modeling, RACF copies the access list exactly. If the creator's user ID appears in the model's access list, RACF copies the authority to the new profile. For example, if the creator's user ID appears in the model's access list with READ, RACF copies that access authority to the new profile without changing it to ALTER as could have occurred in releases prior to OS/390 Release 3.

An important exception for NOADDCREATOR occurs when the user creates a discrete DATASET or TAPEVOL profile using RACROUTE REQUEST=DEFINE. In this case, RACF ignores the NOADDCREATOR options and places the profile creator's user ID on the new profile's access list with ALTER authority. If the profile creator uses profile modeling to define a discrete DATASET or TAPEVOL and the creator's user id appears in the model's access list, RACF creates the authority in the new profile with ALTER authority. This exception to NOADDCREATOR allows system components to allocate data sets and immediately access them without having an administrator manipulate the profile's access list in the interim.

Note: The initial setting of the ADDCREATOR/NOADDCREATOR keyword depends on whether your database is new or old. When IRRMIN00 is run with PARM=NEW, the initial setting is NOADDCREATOR. When IRRMIN00 is run with anything other than PARM=NEW, RACF retains the current value of ADDCREATOR/NOADDCREATOR. For compatibility and migration reasons, this value is set to ADDCREATOR if no prior specification of ADDCREATOR or NOADDCREATOR had occurred.

ADSP | NOADSP**ADSP**

specifies that data sets created by users who have the automatic data set protection (ADSP) attribute is RACF-protected automatically.

ADSP is in effect when RACF is using a newly initialized database.

Because ADSP forces the creation of a discrete profile for each data set created by users who have the ADSP attribute, you should normally specify NOADSP if you specify GENERIC.

NOADSP

cancels automatic RACF protection for users who have the ADSP attribute.

Because ADSP forces the creation of a discrete profile for each data set created by users who have the ADSP attribute, you should normally specify NOADSP if you specify GENERIC.

APPLAUDIT | NOAPPLAUDIT**APPLAUDIT**

specifies that auditing of APPC transactions on your system be enabled. APPC transactions are audited when they receive authorization (start) or have authorization removed (end). You must request auditing for the appropriate APPL profile. Otherwise, turning APPLAUDIT on does not cause

SETROPTS

auditing of APPC transactions. See *z/OS Security Server RACF Auditor's Guide* for more information on requesting auditing.

You must have the AUDITOR attribute to specify this option.

NOAPPLAUDIT

specifies that auditing of APPC transactions on your system (starting and ending) be disabled. You must have the AUDITOR attribute to specify this option.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([*node*].*userid* ...)

specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

ONLYAT([*node*].*userid* ...)

specifies that the command is to be directed only to the node specified by *node* where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed only to the local node.

Note: SETROPTS LIST with no other keywords specified is not eligible for automatic command direction. Do not specify the ONLYAT and LIST keywords together without any other keywords on a SETROPTS command.

AUDIT | NOAUDIT

AUDIT(*class-name* ... | *)

specifies the names of the classes for which you want RACF to perform auditing. For the classes you specify, RACF logs all uses of the RACROUTE REQUEST=DEFINE SVC and all changes made to profiles by RACF commands. (RACF adds the classes you specify to those already specified for auditing.)

The valid class names are USER, GROUP, DATASET, and those defined in the class descriptor table. For a list of general resource classes supplied by IBM, see Appendix B, "Description of RACF Classes" on page 507.

If you specify an asterisk (*), logging occurs for all classes.

You must have the AUDITOR attribute to enter the AUDIT operand.

Note: If you activate auditing for a class using SETROPTS AUDIT, RACF activates auditing for all classes in the class descriptor table that have the same POSIT value as the class you specify. For example, the classes TIMS, GIMS, and AIMS all have a POSIT value of 4 in their respective class descriptor table entries. If you activate auditing for any one of these classes, you activate auditing for all of them.

For more information, see the description of the ICHERCDE macro in *z/OS Security Server RACF Macros and Interfaces*.

NOAUDIT(*class-name* ... | *)

specifies the names of the classes for which you no longer want RACF to

perform auditing. For the classes you specify, RACF no longer logs all uses of the REQUEST=DEFINE SVC and all changes made to profiles by RACF commands. The valid class names are USER, GROUP, DATASET, and those classes defined in the class descriptor table. For a list of general resource classes supplied by IBM, see Appendix B, “Description of RACF Classes” on page 507.

NOAUDIT(*) is in effect when RACF is using a newly-initialized database. If you specify an asterisk (*), logging occurs for any of the classes.

You must have the AUDITOR attribute to enter the NOAUDIT operand.

Note: If you deactivate auditing for a class using SETROPTS NOAUDIT, RACF deactivates auditing for all classes in the class descriptor table that have the same POSIT value as the class you specify. For example, the classes TIMS, GIMS, and AIMS all have a POSIT value of 4 in their respective class descriptor table entries. If you deactivate auditing for any one of these classes, you deactivate auditing for all of them.

For more information, see the description of the ICHERCDE macro in *z/OS Security Server RACF Macros and Interfaces*.

CATDSNS | NOCATDSNS

CATDSNS (FAILURES | WARNING)

specifies that uncataloged data sets, new (and not cataloged), or system temporary data sets are not to be accessed by users.

The following exceptions apply:

1. The job that creates the data set can access it even if the data set is uncataloged. If the data set is still uncataloged when the job ends, it is inaccessible thereafter.
2. Data sets with discrete profiles can be accessed—even if uncataloged—if allowed by the profile.
3. For data sets that have no discrete profile, if private catalogs for the job are in use (STEP CAT and JOB CAT statements), RACF checks the user's authority to a resource named ICHUSERCAT in the FACILITY class. If the resource is protected and the user has access to it, processing continues with the next step. Otherwise, access to the data set is denied.
4. For uncataloged data sets without discrete profiles, RACF constructs a resource name of ICHUNCAT.*dsname* (only the first 30 characters of the *dsname* is used). It checks the user's authority to this resource in the FACILITY class. If the resource is protected by a FACILITY class profile, and the user has access to it, the access is allowed.
5. If the user has the SPECIAL attribute, the access is allowed even if the data set is uncataloged, but a warning message and SMF record is created.
6. Write requests to tape data sets are denied because of SETROPTS CATDSNS.

CATDSNS might have a negative impact on RACF and system performance because RACF must verify that data sets are cataloged before it allows them to be opened.

SETROPTS

Note: For additional information about accessing uncataloged data sets, refer to SETROPTS command in *z/OS Security Server RACF Security Administrator's Guide*.

FAILURES

specifies that RACF is to reject any request to access a data set that is not cataloged.

FAILURES is the default.

If CATDSNS(FAILURES) is in effect and a privileged started task or a user with the SPECIAL attribute requests access of an uncataloged data set, RACF accepts the request and issues a warning message.

WARNING

specifies that the access is allowed even if the data set is uncataloged. However, a warning message and SMF record is created.

NOCATDSNS

specifies that data sets that are not cataloged can be accessed by users.

NOCATDSNS is in effect when RACF is using a newly-initialized database.

CLASSACT | NOCLASSACT

CLASSACT(*class-name ...|**)

specifies those classes defined by entries in the class descriptor table for which RACF protection is to be in effect.

If you specify an asterisk (*), you activate RACF protection for all classes defined in the class descriptor table except for those classes with a default return code of 8. For a list of general resource classes supplied by IBM, see Appendix B, "Description of RACF Classes" on page 507.

Notes:

1. If you activate a class using SETROPTS CLASSACT, RACF activates all classes in the class descriptor table that have the same POSIT value as the class you specify. For example, the classes TIMS, GIMS, and AIMS all have a POSIT value of 4 in their respective class descriptor table entries. If you activate any one of these classes, you activate all of them.

For more information, see the description of the ICHERCDE macro in *z/OS Security Server RACF Macros and Interfaces*.

2. Before activating a class that has a default return code of 8 in the class descriptor table (either explicitly or by means of a shared POSIT value), be sure you have defined the necessary profiles to allow your users to access resources in that class. For example, if you activate JESINPUT without defining profiles to allow access, no one is able to submit batch jobs.

NOCLASSACT(*class-name|**)

specifies those classes defined by entries in the class descriptor table for which RACF protection is not to be in effect. If you specify an asterisk (*), you deactivate RACF protection for all classes defined in the class descriptor table. For a list of general resource classes supplied by IBM, see Appendix B, "Description of RACF Classes" on page 507.

NOCLASSACT is in effect when RACF is using a newly-initialized database.

Rules:

- If you deactivate a class using SETROPTS NOCLASSACT, RACF deactivates all classes in the class descriptor table that have the same POSIT value as the class you specify. For example, the classes TIMS, GIMS, and AIMS all have a POSIT value of 4 in their respective class descriptor table entries. If you deactivate any one of these classes, you deactivate all of them.
- If MACTIVE or MLS is active, you may not deactivate SECLABEL, issuing SETROPTS NOCLASSACT(SECLABEL) will fail.

For more information, see the description of the ICHERCDE macro in *z/OS Security Server RACF Macros and Interfaces*.

CMDVIOL | NOCMDVIOL

specify whether RACF is to log violations detected by RACF commands. You must have the AUDITOR attribute to specify these options.

CMDVIOL

specifies that RACF is to log violations detected by RACF commands (except LISTDSD, LISTGRP, LISTUSER, RLIST, and SEARCH) during RACF command processing. A violation might occur because a user is not authorized to modify a particular profile or is not authorized to enter a particular operand on a command.

CMDVIOL is in effect when RACF is using a newly-initialized database.

NOCMDVIOL

specifies that RACF is not to log violations detected by RACF commands during RACF command processing (except RVARY and SETROPTS, which are always logged).

COMPATMODE | NOCOMPATMODE**COMPATMODE**

allows users and jobs not using SECLABELs to be on a system enforcing SECLABELs. The ACEE's of the user IDs or jobs must have been created by a RACROUTE REQUEST=VERIFY that did not specify RACF Release 1.9 or later keywords.

NOCOMPATMODE

Users and jobs must be running with correct security labels to access data.

NOCOMPATMODE is in effect when RACF is using a newly-initialized database.

EGN | NOEGN

activate or deactivate enhanced generic naming (EGN).

EGN

activates EGN. When you activate this option, RACF allows you to specify the generic character ** (in addition to the generic characters * and %) when you define data set profile names and entries in the global access checking table.

Notes:

1. EGN changes the meaning of the generic character *.
2. When you first activate enhanced generic naming, the RACF-protection provided by existing data set profiles and global access checking table remains the same.

SETROPTS

For information on EGN and its effect on profile names, see the description of generic profiles in Appendix A, “Naming Considerations for Resource Profiles”.

NOEGN

specifies deactivation of EGN. When you deactivate this option, RACF does not allow you to specify the generic character ** when you define data set names and entries in the global access checking table.

NOEGN is in effect when RACF is using a newly-initialized database.

Attention

If you protect data sets with generic profiles while EGN is active and then deactivate this option, your resources can no longer be protected. Table 81 on page 499 and Table 82 on page 499 show examples of generic profiles created with enhanced generic naming active.

Some of these profiles do not provide RACF protection when the option is deactivated. If a data set is unprotected when EGN is deactivated, you can protect the data set with a discrete profile as described in Appendix A, “Naming Considerations for Resource Profiles” either before or after the option is deactivated, or with a generic profile after the option is deactivated.

EIMREGISTRY | NOEIMREGISTRY

The keyword EIMREGISTRY makes the name of the RACF registry available to the EIM services. It is used by invokers of EIM services that choose not to specify a registry name on an EIM domain lookup operation. Applications or other services that use the EIM services may instruct their invokers to define the local registry name in the IRR.PROXY.DEFAULTS FACILITY class profile and activate it using SETROPTS EIMREGISTRY.

Changes to the IRR.PROXY.DEFAULTS profile are not reflected in the in-storage copy until SETROPTS EIMREGISTRY is issued or the system is IPLed.

EIMREGISTRY

Activates the registry name defined in the IRR.PROXY.DEFAULTS FACILITY class profile.

NOEIMREGISTRY

Deactivates the registry name defined in the IRR.PROXY.DEFAULTS FACILITY class profile. The registry name in the IRR.PROXY.DEFAULTS FACILITY class profile must be deleted using the RALTER FACILITY IRR.PROXY.DEFAULTS EIM(NOLOCALREGISTRY) command in order to make this deletion permanent across IPLs.

ERASE | NOERASE

ERASE(*erase-indicator*)

specifies that data management is to physically erase the DASD data set extents at the time the DASD data set is deleted (scratched) or released for reuse. When RACF is running on a system that includes data management support for erase-on-scratch, the allocated extents of a scratched and erased data set are overwritten with binary zeros.

If you specify ERASE without any suboperands, whether a scratched data set is erased depends on the contents of the erase indicator in the data set profile. The ERASE suboperands allow you to override the erase indicator

in the data set profile, to control the scope of erase-on-scratch on an installation level rather than leaving it to individual users.

The variable *erase-indicator* can be:

ALL

specifies that data management is to erase all scratched DASD data sets, including temporary data sets, regardless of the erase indicator, if any, in the data set profile.

SECLEVEL(*secllevel-name*)

specifies that data management is to erase all scratched DASD data sets that have a security level equal to or greater than the security level that you specify, where *secllevel-name* must be a member of the SECLEVEL profile in the SECDATA class.

Note: Scratched DASD data sets with a security level lower than the level you specify is not erased, regardless of the erase indicators (if any) in the data set profiles.

NOSECLEVEL

specifies that RACF is not to use the security level in the data set profile when it decides whether data management is to erase a scratched DASD data set.

Specifying ERASE(NOSECLEVEL) causes RACF to use the erase indicator in the data set profile to decide whether data management is to scratch the data set. NOSECLEVEL is the default if you do not specify *erase-indicator* when you specify ERASE.

NOERASE

specifies that erase-on-scratch processing is not in effect. NOERASE means that no DASD data sets are erased when deleted (scratched), even if the erase indicator in the data set profile is on.

NOERASE is in effect when RACF is using a newly-initialized database.

GENCMD | NOGENCMD

GENCMD(*class-name ...I)**

activates generic profile command processing for the specified classes. Valid class names you can specify are DATASET and all class names defined in the class descriptor table except grouping classes. If you specify an asterisk (*), you activate generic profile command processing for the DATASET class plus all the classes defined in the class descriptor table except grouping classes. For a list of general resource classes supplied by IBM, see Appendix B, "Description of RACF Classes".

When GENCMD is in effect for a class, all the command processors can work on generic profiles, but the RACF SVC routines cannot perform generic profile checking. This operand allows the installation to temporarily disable generic profile checking (during maintenance, for example) and still use the RACF commands to maintain generic profiles.

Note: If you activate generic profile command processing for a class using SETROPTS GENCMD, RACF activates generic profile command processing for all classes in the class descriptor table that have the same POSIT value as the class you specify, except grouping classes. For example, the resource classes TIMS and AIMS and the grouping class GIMS all have a POSIT value of 4 in their respective class descriptor table entries. If you activate generic profile

SETROPTS

command processing for TIMS, you also activate it for AIMS. However, you cannot activate this option for GIMS because GIMS is a grouping class.

For more information, see the description of the ICHERCDE macro in *z/OS Security Server RACF Macros and Interfaces*.

NOGENCMD(*class-name ...|**)

deactivates generic profile command processing for the specified classes. Valid class names you can specify are DATASET and all class names defined in the class descriptor table except grouping classes. If you specify an asterisk (*), you deactivate generic profile command processing for the DATASET class plus all the classes in the class descriptor table (excluding grouping classes). For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, "Description of RACF Classes" on page 507.

NOGENCMD(*) is in effect when RACF is using a newly-initialized database.

If generic profile checking is active (GENERIC is in effect), RACF ignores this operand because GENERIC both includes and overrides generic profile command processing.

Note: If you deactivate generic profile command processing for a class using SETROPTS NOGENCMD, RACF deactivates generic profile command processing for all classes in the class descriptor table that have the same POSIT value as the class you specify, except grouping classes. For example, the resource classes TIMS and AIMS and the grouping class GIMS all have a POSIT value of 4 in their respective class descriptor table entries. If you deactivate generic profile command processing for TIMS, you also deactivate it for AIMS. However, GIMS is unaffected because it is a grouping class.

For more information, see the description of the ICHERCDE macro in *z/OS Security Server RACF Macros and Interfaces*.

GENERIC | NOGENERIC

GENERIC(*class-name ...|**)

activates generic profile checking for the classes specified. Valid class names are DATASET and all class names defined in the class descriptor table except grouping classes. If you specify an asterisk (*), you activate generic profile checking for the DATASET class plus all the classes in the class descriptor table except grouping classes. For a list of general resource classes supplied by IBM, see Appendix B, "Description of RACF Classes" on page 507.

Generic profile command processing is automatically activated for all classes for which generic profile checking is activated. Note that GENCMD must be explicitly used.

If you specify GENERIC with REFRESH, only those currently active and authorized classes are refreshed.

Notes:

1. If RACF is enabled for sysplex communication, RACF propagates SETROPTS GENERIC(*class-name*) REFRESH commands to other members of the data sharing group.

2. If RACF is not enabled for sysplex communication, a SETROPTS GENERIC(*class-name*) REFRESH command is effective only on the system where it is issued.
3. If you specify GENERIC, you should also specify NOADSP.
4. If you activate generic profile checking for a class using SETROPTS GENERIC, RACF activates generic profile checking for all classes in the class descriptor table that have the same POSIT value as the class you specify, except grouping classes. For example, the resource classes TIMS and AIMS and the grouping class GIMS all have a POSIT value of 4 in their respective class descriptor table entries. If you activate generic profile checking for TIMS, you also activate it for AIMS. However, you cannot activate this option for GIMS because GIMS is a grouping class.

For more information, see the description of the ICHERCDE macro in *z/OS Security Server RACF Macros and Interfaces*.

NOGENERIC(*class-name ...!)**

deactivates the generic profile checking facility for the classes specified. Valid class names you can specify are DATASET and all class names defined in the class descriptor table except grouping classes. If you specify an asterisk (*), you deactivate generic profile checking for the DATASET class plus all the classes in the class descriptor table (excluding grouping classes). For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, “Description of RACF Classes” on page 507.

NOGENERIC (*) is in effect when RACF is using a newly-initialized database.

NOGENERIC does not automatically deactivate generic profile command processing. Note that NOGENCMD must be explicitly used.

If you specify GENCMD with NOGENERIC, users can issue RACF commands to maintain generic profiles, but RACF does not use generic profile checking during authorization checking.

If you specify NOGENCMD with NOGENERIC, all generic profile command processing is deactivated.

Note: If you deactivate generic profile checking for a class using SETROPTS NOGENERIC, RACF deactivates generic profile checking for all classes in the class descriptor table that have the same POSIT value as the class you specify, except grouping classes. For example, the resource classes TIMS and AIMS and the grouping class GIMS all have a POSIT value of 4 in their respective class descriptor table entries. If you deactivate generic profile checking for TIMS, you also deactivate it for AIMS. However, GIMS is unaffected because it is a grouping class.

For more information, see the description of the ICHERCDE macro in *z/OS Security Server RACF Macros and Interfaces*.

GENERICOWNER | NOGENERICOWNER

GENERICOWNER

restricts creation of profiles in all general resource classes except the PROGRAM class.

SETROPTS

To create a profile that is more specific than any existing profile protecting the same resource a user must:

- Have the SPECIAL attribute
- Be the owner of the existing profile
- Have the group-SPECIAL attribute if a group owns the profile
- Have the group-SPECIAL attribute if the owner of the profile is in the scope of the group

Notes:

1. GENERICOWNER provides protection only when there is an existing (less-specific) profile protecting the resource.
2. A less-specific profile must end in * or ** or %(s). A more-specific profile is a profile that matches the less-specific profile name, character for character, up to the ending * or ** or %(s) in the less-specific name. If the less-specific profile ends in %(s), the characters in the more-specific profile that correspond to the contiguous trailing %(s) must NOT be either * or period. For more information, see “Permitting Profiles for GENERICOWNER Classes” on page 503.

For example: To allow USERX to RDEFINE A.B in the JESSPOOL class, you need profile A.* in the JESSPOOL class, which is owned by USERX. You also need profile **, owned by the system administrator, to prevent other CLAUTH users from being able to RDEFINE A.B.

3. GENERICOWNER does not prevent the creation of a more specific profile if the more specific profile is created in the grouping class and is specified on the ADDMEM operand. For example, profile A* exists in the TERMINAL class and is owned by a group for which user ELAINE does not have group-SPECIAL. If the GENERICOWNER option is in effect, user ELAINE cannot define a more specific profile in the member class (such as, RDEF TERMINAL AA*), but user ELAINE can define a profile if it is specified on the ADDMEM operand for the grouping class profile (such as, RDEF GTERMINL profile-name ADDMEM(AA*)).

NOGENERICOWNER

cancels the restriction on the creation of profiles for general resources.

NOGENERICOWNER is in effect when RACF is using a newly-initialized database.

GENLIST | NOGENLIST

GENLIST(class-name ...)

Also see RACLIST operand.

Activates the sharing of in-storage generic profiles for the classes specified. When GENLIST is active for a class, the generic profiles for that class are loaded into common storage (ECSA) instead of being resident in the private storage (ELSQA) of each user who references the class. Before activating GENLIST for a class, you should check with your system programmer to determine if your system is configured with enough ECSA to contain the profiles. The *z/OS Security Server RACF System Programmer's Guide* contains information about the amount of virtual storage required for generic profiles.

The following classes supplied by IBM can be used with GENLIST:

APPL	FACILITY	KEYSMSTR	TERMINAL	VMMDISK
CPSMOBJ	FIELD	LOGSTRM	TMEADMIN	VMNODE

DASDVOL	INFOMAN	RRSFDATA	VMBATCH	VMRDR
DCEUIDS	JESJOBS	SDSF	VMCMD	VMSEGMT
DSNR				

When you activate GENLIST processing for a class, a generic profile in that class is copied from the RACF database into common storage the first time an authorized user requests access to a resource protected by the profile. The profile is retained in common storage and is available for all authorized users, thus saving real storage because the need to retain multiple copies of the same profile (one copy for each requesting user) in common storage is eliminated. Also, because RACF does not have to retrieve the profile each time a user requests access to a resource protected by it, this function saves processing overhead.

If you want to refresh shared in-storage generic profiles for a specific resource class, issue the SETROPTS command with the `GENERIC(class-name)` and `REFRESH` operands.

Note: RACF does not allow you to specify SETROPTS GENLIST and SETROPTS RACLIST for the same general resource class.

NOGENLIST(*class-name ...*)

Also see NORACLIST operand.

Deactivates the sharing of in-storage generic profiles for the classes specified. Deactivate this function for general resource classes defined in the class descriptor table that are eligible for GENLIST processing. These classes are listed under the description for GENLIST.

When you specify NOGENLIST, RACF deletes in-storage generic profiles for the specified classes from common storage.

NOGENLIST is in effect for all classes defined in the class descriptor table when RACF is using a newly-initialized database.

GLOBAL | NOGLOBAL

GLOBAL(*class-name ...|**)

Specifies those classes eligible for global access checking. If you specify an asterisk (*), you activate global access checking for all valid classes.

Valid classes you may specify are:

- The DATASET class
- The NODE grouping class
- The SECLABEL grouping class
- All other classes defined in the class descriptor table, except for the remaining grouping classes

For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, “Description of RACF Classes” on page 507.

If you specify GLOBAL with REFRESH, only those currently active and authorized classes are refreshed. If you have deleted the GLOBAL profile for a class, you should issue the SETROPTS command with the NOGLOBAL operand specified, rather than GLOBAL with REFRESH specified.

SETROPTS

Notes:

1. If you activate global access checking for a class using SETROPTS GLOBAL, RACF activates global access checking for all classes in the class descriptor table that have the same POSIT value as the class you specify, except the excluded grouping classes. For example, the resource classes TIMS and AIMS and the grouping class GIMS all have a POSIT value of 4 in their respective class descriptor table entries. If you activate global access checking for TIMS, you also activate it for AIMS. However, you cannot activate this option for GIMS because GIMS is a grouping class.

For more information, see the description of the ICHERCDE macro in *z/OS Security Server RACF Macros and Interfaces*.
2. If RACF is enabled for sysplex communication, it propagates the SETROPTS GLOBAL and SETROPTS GLOBAL REFRESH commands to other systems in the sysplex if the command is successful on the system on which it was entered. If RACF is not enabled for sysplex communication, the command has to be issued on each system sharing the database.
3. Global access checking is bypassed if the user ID has the RESTRICTED attribute.

NOGLOBAL(*class-name ...*!*)

Deactivates global access checking for the specified classes. For more information on classes that are process by the NOGLOBAL operand, see the GLOBAL operand description.

NOGLOBAL(*) is in effect when RACF is using a newly-initialized database.

Note: If you deactivate global access checking for a class using SETROPTS NOGLOBAL, RACF deactivates global access checking for all classes in the class descriptor table that have the same POSIT value as the class you specify, except for the excluded grouping classes. For example, the resource classes TIMS and AIMS and the grouping class GIMS all have a POSIT value of 4 in their respective class descriptor table entries. If you deactivate global access checking for TIMS, you also deactivate it for AIMS. However, GIMS is unaffected because it is a grouping class.

For more information, see the description of the ICHERCDE macro in *z/OS Security Server RACF Macros and Interfaces*.

GRPLIST | NOGRPLIST

GRPLIST

specifies that authorization checking processing is to perform list-of-groups access checking for all system users. When you specify GRPLIST, a user's authority to access or define a resource is not based only on the authority of the user's current connect group; access is based on the authority of any group to which the user is connected.

NOGRPLIST

specifies that the user's authority to access a resource is based on the authority of the user's current connect group.

NOGRPLIST is in effect when RACF is using a newly-initialized database.

INACTIVE | NOINACTIVE

INACTIVE(*unused-userid-interval*)

specifies the number of days (1 to 255) that a user ID can remain unused and still be considered valid. RACF user verification checks the number of days since the last successful time the user accessed the system against the INACTIVE value and, if the former is larger, revokes the user's right to use the system. If you specify INACTIVE, INITSTATS must be in effect.

If the backup database is needed but does not contain current information, some user IDs can be revoked because they appear to have been unused beyond the number of days specified on the INACTIVE operand. For more information, see *z/OS Security Server RACF System Programmer's Guide*.

NOINACTIVE

specifies that RACF user verification is not to check user IDs against an *unused-userid-interval*.

NOINACTIVE is in effect when RACF is using a newly-initialized database.

INITSTATS | NOINITSTATS**INITSTATS**

specifies that statistics available during RACF user verification are to be recorded. These statistics include the date and time the user was verified by RACF, the number of user verifications that specified a particular group, and the date and time of the user last requested verification with a particular group. If you specify INACTIVE, REVOKE, HISTORY, or WARNING, INITSTATS must be in effect.

INITSTATS is in effect when RACF is using a newly-initialized database.

NOINITSTATS

specifies that statistics available during user verification are not to be recorded.

JES

controls job entry subsystem (JES) options. The JES options are:

BATCHALLRACF | NOBATCHALLRACF**BATCHALLRACF**

specifies that JES is to test for the presence of a user ID and password on the job statement or for propagated RACF identification information for all batch jobs. If the test fails, JES is to fail the job.

NOBATCHALLRACF

specifies that JES is not to test for the presence of a user ID and a password on the statement, or propagated RACF identification information for all batch jobs.

NOBATCHALLRACF is in effect when RACF is using a newly-initialized database.

EARLYVERIFY | NOEARLYVERIFY**EARLYVERIFY**

specifies that JES is to invoke the system authorization facility (SAF) for jobs that do not qualify for user identification propagation. SAF can call an installation-written exit routine (if installed) for further verification of the user ID, group, and password (if specified) at job submission time. See *z/OS Security Server RACROUTE Macro Reference* for further information about the MVS router exit.

SETOPTS

NOEARLYVERIFY

specifies that the RACF CVT indicator is not to be set and SAF is not to get control.

NOEARLYVERIFY is in effect when RACF is using a newly-initialized database.

NOEARLYVERIFY has no effect if you are using JES 3.1.3 or later.

XBMALLRACF | NOXBMALLRACF

XBMALLRACF

specifies that JES is to test for the presence of either a user ID and password on the JOB statement, or JES-propagated RACF identification information for all jobs to be run with an execution batch monitor. If the test fails, JES is to fail the job.

XBMALLRACF is only used on JES2.

NOXBMALLRACF

specifies that JES is not to test for the presence of either a user ID and password on the JOB statement, or JES-propagated RACF identification information for all jobs to be run with an execution batch monitor.

NOXBMALLRACF is in effect when RACF is using a newly-initialized database.

NJUSERID(*userid*)

defines the name (user ID) associated with SYSOUT or jobs that arrive through the network without an RTOKEN or UTOKEN.

The initial user ID (default user ID) after RACF data set initialization is ???????? (eight question marks).

Note: The variable *userid* cannot be a user ID defined in the RACF database. For more information, see the section on providing security for JES in *z/OS Security Server RACF Security Administrator's Guide*.

UNDEFINEDUSER(*userid*)

defines the name (user ID) that is associated with local jobs that enter the system without a user ID.

The initial user ID (default user ID) after RACF data set initialization is ++++++ (eight plus signs).

Note: The variable *userid* cannot be a user ID defined in the RACF database. For more information, see the section on providing security for JES in *z/OS Security Server RACF Security Administrator's Guide*.

[KERBLVL(0|1)]

specifies what level of encryption processing should occur when a KERB segment is being processed for user and general resource profiles.

- 0** specifies that only DES keys should be created and the ENCTYPE fields will be set to X'00000001'. The value of the ENCRYPT field will be ignored by R_kerbinfo.
- 1** specifies that keys should be created for DES, DES3 and DESD. This level also allows for the enablement and disablement of these key types for a specific profile through the use of the ENCRYPT field.

Note: If the value of KERBLVL is reset from a value of 1 to 0, some processing may not perform as expected. When this is done the listing of profiles that had DES3 or DESD encryption enabled or, DES excryption disabled will still show this as is denoted by the profile settings. The IBM Kerberos server (through the use of the R_kerbinfocall) will process as if DES is enabled and the others are disabled, regardless of whether the profile is defined to have these encryption types enabled or disabled. Setting KERBLVL to a value lower than the current value should be avoided due to this inconsistency.

LANGUAGE

specifies the system-wide defaults for national languages (such as American English or Japanese) to be used on your system. You can specify a primary language, a secondary language, or both. The languages you specify depend on which products, when installed on your system, check for primary and secondary languages (using RACROUTE REQUEST=EXTRACT):

- If this user establishes an extended MCS console session, the languages you specify should be the same as the languages specified on the LANGUAGE LANGCODE statements in the MMSLSTxx PARMLIB member. See your MVS system programmer for this information.
- If this is a CICS user, see your CICS administrator for the languages supported by CICS on your system.

The SETROPTS LANGUAGE operand does not affect the language in which the RACF ISPF panels are displayed. The order in which the RACF ISPF panel libraries are allocated determines the language used. If your installation ordered a translated feature of RACF, the RACF program directory gives instructions for setting up the ISPF panels.

PRIMARY(*language*)

specifies the installation's default primary language.

The variable *language* can be a quoted or unquoted string.

If the PRIMARY suboperand is not specified, the primary language is not changed.

SECONDARY(*language*)

specifies the installation's default secondary language.

The language name can be a quoted or unquoted string.

If the SECONDARY suboperand is not specified, the secondary language is not changed.

Notes:

1. For both the PRIMARY and SECONDARY suboperands, specify the installation-defined name of a currently active language (a maximum of 24 characters) or one of the language codes (3 characters in length) that is installed on your system. For a list of valid codes, see *National Language Design Guide, National Language Support Reference Manual*, SE09-8002.
2. If the MVS message service is not active, the PRIMARY and SECONDARY values must be a 3 character language code.
3. The same language can be specified for both PRIMARY and SECONDARY.
4. RACF is shipped with both the primary and secondary language defaults set to ENU, meaning United States English.

SETROPTS

LIST

specifies that the current RACF options are to be displayed. If you specify operands in addition to LIST on the SETROPTS command, RACF processes the other operands before it displays the current set of options.

If RACF is enabled for sysplex communication and the system is in read-only mode, users on that system can issue the SETROPTS LIST command. All other operands are ignored.

You must have the SPECIAL, AUDITOR, group-SPECIAL, or group-AUDITOR attribute to enter the LIST operand.

If you have the SPECIAL or group-SPECIAL attribute, RACF displays all operands except these auditing operands:

- APPLAUDIT | NOAPPLAUDIT
- AUDIT | NOAUDIT
- CMDVIOL | NOCMDVIOL
- LOGOPTIONS
- OPERAUDIT | NOOPERAUDIT
- SAUDIT | NOSAUDIT
- SECLABELAUDIT | NOSECLABELAUDIT.

If you have the AUDITOR or the group-AUDITOR attribute, RACF displays all operands.

Note: SETROPTS LIST with no other keywords specified is not eligible for automatic command direction. Do not specify the ONLYAT and LIST keywords together without any other keywords on a SETROPTS command.

LOGOPTIONS (*auditing-level (class-name...) ...*)

audits access attempts to resources in specified classes according to the auditing-level specified. You must have the AUDITOR attribute. You can specify the DATASET class and any classes in the class descriptor table. The resources need not have profiles created in order for auditing to occur. See *z/OS Security Server RACF Auditor's Guide* for more information on when auditing occurs.

The SUCSESSES and FAILURES operands result in auditing in addition to any auditing specified in profiles in the class. In contrast, the ALWAYS and NEVER operands override any auditing specified in profiles in the class. Note that LOG=NONE, specified on a RACROUTE REQUEST=AUTH, takes precedence (auditing is not performed).

auditing-level

specifies the access attempts to be logged for *class-name*. These options are processed in the order listed below. Thus, if *class-name* is specified with both SUCSESSES and ALWAYS in the same command, auditing takes place at the SUCSESSES level because option SUCSESSES is processed after ALWAYS.

ALWAYS All access attempts to resources protected by the class are audited.

NEVER No access attempts to resources protected by the class are audited. (All auditing is suppressed.)

SUCSESSES All successful access attempts to resources protected by the class are audited.

FAILURES	All failed access attempts to resources protected by the class are audited.
DEFAULT	Auditing is controlled by the profile protecting the resource, if a profile exists. You can specify DEFAULT for all classes by specifying an asterisk (*) with DEFAULT.

LOGOPTIONS(DEFAULT) is in effect when RACF is using a newly-initialized database.

class-name

the RACF class to which *auditing-level* applies. *Class-name* can be DATASET and any classes in the class descriptor table. Each class can have only one auditing level associated with it. The auditing-levels are processed in the following order:

1. ALWAYS
2. NEVER
3. SUCCESSES
4. FAILURES
5. DEFAULT

This processing order occurs independently of the order you specify the auditing-levels. If you specify two or more auditing-levels for a class in the same command, *only the last option processed takes effect*. Thus, if you specify the following command:

```
SETR LOGOPTIONS ( FAILURES ( DATASET, SECLABEL ),
                  ALWAYS ( DATASET, APPL ),
                  DEFAULT (DATASET, GLOBAL ) )
```

The options in effect for the classes is:

- ALWAYS — APPL
- FAILURES — SECLABEL
- DEFAULT — DATASET, GLOBAL

Classes DATASET and APPL are first assigned auditing-level ALWAYS. Class DATASET is then assigned auditing-level FAILURES, as is class SECLABEL. Finally, class DATASET is assigned DEFAULT auditing-level, as is class GLOBAL.

If you specify one *auditing-level* for *class-name* and in a separate command specify a new auditing level for the same class name, the new auditing level take effects.

SETROPTS LOGOPTIONS(DEFAULT(*)) is the equivalent to previous versions of RACF. It is in effect when RACF is using a newly-initialized database.

MLACTIVE | NOMLACTIVE

For the relationships among SECLABEL, MLS, MACTIVE, and MLQUIET, see *z/OS Security Server RACF Security Administrator's Guide*.

MLACTIVE (FAILURES | WARNING)

causes security labels to be required on all work entering the system and on all resources defined to USER, DATASET, and all classes defined in the class descriptor table that require SECLABEL.

Rules:

SETROPTS

- This option is available only if the SECLABEL class is active.
- Activation of MACTIVE will fail if the SECLABEL class is not active or being activated by the command activating MACTIVE.

FAILURES

specifies that RACF is to reject any request to create or access any resource which requires a SECLABEL in the profile that protects it, and does not have one, and to reject any work entering the system which does not have a SECLABEL.

The only exception is if MLS(FAILURES) and MACTIVE(FAILURES) are in effect, and a privileged started task or a user with the SPECIAL attribute and the SYSHIGH SECLABEL attempts to access a resource that requires a SECLABEL and does not have one. In this case, RACF allows the request as long as the request does not declassify data.

MLACTIVE(FAILURES) is the default value.

WARNING

specifies that when a user requests access to a resource that does not have a SECLABEL and the resource belongs to a class that requires SECLABELs, access is allowed but a warning is issued. Also, when work enters the system without a SECLABEL, access is allowed but a warning is issued.

NOMLACTIVE

allows work to enter the system without a SECLABEL and allows requests to access a resource that does not have a SECLABEL and the resource belongs to a class that requires SECLABELs.

NOMLACTIVE is in effect when RACF is using a newly-initialized database.

MLQUIET | NOMLQUIET

For the relationships among SECLABEL, MLS, MACTIVE, and MLQUIET, see *z/OS Security Server RACF Security Administrator's Guide*.

MLQUIET

allows only started tasks, console operators, or users with the SPECIAL attribute to log on, start new jobs, or access resources. Actions requiring user verification, resource access checking, or resource definition are available only to the security administrator (SPECIAL user), a trusted computer base job (as indicated in the token), or the console operator.

When this option is enabled, the system is in a tranquil state.

NOMLQUIET

allows all users access to the system.

NOMLQUIET is in effect when RACF is using a newly-initialized database.

MLS | NOMLS

For the relationships among SECLABEL, MLS, MACTIVE, and MLQUIET, see *z/OS Security Server RACF Security Administrator's Guide*.

MLS (FAILURES | WARNING)

prevents a user from declassifying data. In order to copy data, the SECLABEL of the target must encompass the SECLABEL of the source.

Rules:

- This option is available only if the SECLABEL class is active.
- Activation of MLS will fail if the SECLABEL class is not active or being activated by the command activating MLS.

FAILURES

specifies that RACF is to reject any request to declassify data.

MLS(FAILURES) is the default value if you do not specify either FAILURES or WARNING.

WARNING

specifies that when a user attempts to declassify data, RACF is to allow the request but issue warning messages to the user and the security administrator.

NOMLS

allows users to declassify data within the same CATEGORY.

NOMLS is in effect when RACF is using a newly-initialized database.

MLSTABLE | NOMLSTABLE**MLSTABLE**

allows the installation to indicate that no one on the system is allowed to alter the SECLABEL of an object or alter the definition of the SECLABEL, unless MLQUIET is in effect.

NOMLSTABLE

allows the alteration of SECLABEL definitions or the SECLABELs within a profile without requiring MLQUIET to be in effect.

NOMLSTABLE is in effect when RACF is using a newly-initialized database.

MODEL | NOMODEL**MODEL**

specifies, through the following suboperands, the model profile processing options. For information about automatic profile modeling, refer to the *z/OS Security Server RACF Security Administrator's Guide*.

GDG | NOGDG

specifies that RACF should attempt to protect RACF-indicated members of a generation data group (GDG) using a base profile with the same name as the GDG data set base name. If a base profile exists for a particular RACF-indicated member, then RACF uses the base profile when determining whether the user can access or create the member. Otherwise, RACF uses, or creates, an individual profile for the model. MODEL(GDG) has no effect on GDG members that are protected by generic profiles.

NOGDG specifies that GDG members should not be treated specially by RACF; they are processed as any other data set would be.

GROUP | NOGROUP

specifies that when creating a new profile for a group-named data set, RACF should check whether a model profile is specified in the group profile. If so, that model profile should be used to complete the definition of the new data set profile.

NOGROUP specifies that RACF should not use model profiles to complete the definition of new group-named data sets.

USER | NOUSER

specifies that when creating a new profile for all user ID-named data sets, RACF should check whether a model profile is specified in the user profile. If so, that model profile should be used to complete the definition of the new data set profile.

SETROPTS

NOUSER specifies that RACF should not use model profiles to complete the definition of new user ID-named data sets.

NOMODEL

specifies that there is no model profile processing for GDG, GROUP, or USER data sets.

NOMODEL is in effect when RACF is using a newly-initialized database.

OPERAUDIT | NOOPERAUDIT

specifies whether RACF is to log all actions allowed only because a user has the OPERATIONS (or group-OPERATIONS) attribute. You must have the AUDITOR attribute to enter these operands.

OPERAUDIT

specifies that RACF is to log all actions, such as accesses to resources and commands, allowed only because a user has the OPERATIONS or group-OPERATIONS attribute.

NOOPERAUDIT

specifies that RACF is not to log the actions allowed only because a user has the OPERATIONS or group-OPERATIONS attribute.

NOOPERAUDIT is in effect when RACF is using a newly-initialized database.

PASSWORD (*suboperands*)

specifies options to monitor and check passwords:

HISTORY | NOHISTORY

HISTORY(*number-previous-passwords*)

specifies the number of previous passwords (1 to 32) that RACF saves for each user ID and compares with an intended new password. If there is a match with one of these previous passwords, or with the current password, RACF rejects the intended new password.

If you increase the password HISTORY number, RACF saves and compares that number of passwords to the new password. If you reduce the password HISTORY number, passwords in the user profile that are beyond the newly specified HISTORY number are never deleted and continue to be used for comparison.

For example, if the HISTORY number is 12 and you reduce that HISTORY number to 8, RACF also compares the old passwords 9 through 12 with the intended new password.

If you specify HISTORY, INITSTATS must be in effect.

NOHISTORY

specifies that new password information is only compared with the current password. If prior password history information exists, it is neither deleted nor changed.

NOHISTORY is in effect when RACF is using a newly-initialized database.

INTERVAL(*password-change-interval*)

specifies the maximum number of days (1 to 254) that each user's password is valid. The value specified for *password-change-interval* becomes the following:

- A default value for new users defined to RACF through the ADDUSER command.

- An upper limit for users who specify the INTERVAL operand on the PASSWORD command.

When a user logs on to the system, RACF compares the system password interval value with the password interval value specified in the user's profile. RACF uses the lower of the two values to determine if the user's password has expired.

The initial default at RACF initialization is 30 days.

REVOKE | NOREVOKE

REVOKE(*number-invalid-passwords*)

specifies the number (1 to 255) of consecutive incorrect password attempts RACF allows before it revokes the user ID on the next incorrect attempt. If you specify REVOKE, INITSTATS must be in effect.

NOREVOKE

specifies that RACF is to ignore the number of consecutive invalid password attempts.

RULE_{*n*} | NORULE_{*n*} | NORULES

Note: The ISPF panels might be easier to use for password rules.

RULE_{*n*} (LENGTH (*m1:m2*) *content-keyword*(*position*))

specifies an individual syntax rule for new passwords that users specify at logon, on JCL job cards, or on the PASSWORD command. Eight syntax rules are allowed; thus, *n* can range from 1 to 8.

These syntax rules do not apply to:

- Logon passwords specified on the ADDUSER command.
- Logon passwords specified on the ALTUSER command with the PASSWORD operand and with the EXPIRED operand either specified or defaulted.
- Default passwords set by the PASSWORD USER(*userid*) command, which are set to the user's default group name.

These rules apply if the NOEXPIRED operand is used with the PASSWORD operand on the ALTUSER command.

If multiple rules are defined, a password that passes at least one rule is accepted.

LENGTH(*m1:m2*)

specifies the minimum and maximum password lengths to which this particular rule applies (*m2* must be greater than or equal to *m1*). Because RACF allows passwords no longer than 8 alphanumeric characters, the value for *m2* must be less than or equal to 8. If you omit the *m2* value, the rule applies to a password of one length only.

content-keyword(*position*)

specifies the syntax rules for the positions indicated by the LENGTH suboperand. The possible values for *content-keyword* are:

ALPHA

Alphabetic characters and national characters # (X'7B'), \$ (X'5B'), and @ (X'7C')

SETROPTS

ALPHANUM

Alphabetic characters, numeric characters, and the national characters # (X'7B'), \$ (X'5B'), and @ (X'7C') This particular content-keyword requires at least one alphabetic or national character and one numeric character.

VOWEL

Vowel characters, namely 'A', 'E', 'I', 'O', and 'U'

NOVOWEL

Characters that are not vowels, including:

- Alphabetic characters that are consonants, not vowels
- National characters
- Numeric characters

CONSONANT

Nonvowel characters

NUMERIC

Numeric characters

Each *content-keyword* is followed by a position (in the form of *k*, not greater than 8), list of positions (form of *k1,k2,k3...* in any order), or a range (form of *k4:k5*, where *k5* must be greater than or equal to *k4*). See the following example:

```
RULE1(LENGTH(8) CONSONANT(1,3,5:8) NUMERIC(2,4))
```

Syntax rule 1 applies to passwords eight characters in length with consonants in positions 1, 3, 5, 6, 7, and 8 and numbers in positions 2 and 4. Thus the password "B2D2GGDD" obeys Rule1, and "C3PIBOLO" does not.

If the values in the *content-keywords* do not define every position specified by the LENGTH value, the undefined positions can consist of any combination of alphanumeric characters.

NORULE_{*n*}

specifies that RACF is to delete the particular rule identified by *n*.

NORULES

specifies that RACF is to delete all password syntax rules established by the installation.

NORULES is in effect when RACF is using a newly-initialized database.

WARNING(*days-before-password-expires*) | NOWARNING

WARNING

specifies the number of days (1 to 255) before a password expires when RACF is to issue a warning message to a TSO user or to the joblog of a batch job which specified a password.

If you specify a WARNING value that exceeds the INTERVAL value, a warning message is issued at each logon. If you do not want the warning with each logon, specify a value for WARNING that is less than the value you specify for INTERVAL. If you specify WARNING, INITSTATS must be in effect.

NOWARNING

specifies that RACF is not to issue the warning message for password expiration.

NOWARNING is in effect when RACF is using a newly-initialized database.

PREFIX | NOPREFIX**PREFIX(*prefix*)**

activates RACF protection for data sets that have single-qualifier names, and specifies the 1-8 character prefix to be used as the high-level qualifier in the internal form of the names. The variable *prefix* should be a predefined group name, and it must not be the high-level qualifier of any actual data sets in the system.

NOPREFIX

deactivates RACF protection for data sets that have single-level names.

When EGN is active and NOPREFIX is in effect, a data set can be protected with a generic profile of the form ABC.**, where ABC equals the data set name.

NOPREFIX is in effect when RACF is using a newly-initialized database.

PROTECTALL | NOPROTECTALL**PROTECTALL (FAILURES | WARNING)**

activates protect-all processing. When protect-all processing is active, the system automatically rejects any request to create or access a data set that is not RACF-protected. This processing includes DASD data sets, tape data sets, catalogs, and GDG basenames. Temporary data sets that comply with standard MVS temporary data set naming conventions are excluded from protect-all processing.

Note that PROTECTALL requires all data sets to be RACF-protected. This includes tape data sets if your installation specifies the TAPEDSN operand on the SETROPTS command.

In order for protect-all to work effectively, you must specify GENERIC to activate generic profile checking. Otherwise, RACF would allow users to create or access only data sets protected by discrete profiles. If your installation uses nonstandard names for temporary data sets, you must also predefine entries in the global access checking table that allow these data sets to be created and accessed.

The WARNING suboperand enables you to specify a warning message to the requestor in place of rejecting the request.

FAILURES

specifies that RACF is to reject any request to create or access a data set that is not RACF-protected.

The default value is FAILURES.

If PROTECTALL(FAILURES) is in effect and a user with the SPECIAL attribute requests access to an unprotected data set, RACF accepts the request, audits the event, and issues a protect-all warning message.

If PROTECTALL(FAILURES) is in effect and a trusted started task requests access to an unprotected data set, RACF accepts the request, audits the event, and no warning message is issued.

SETROPTS

If PROTECTALL(FAILURES) is in effect and a privileged started task requests access to an unprotected data set, RACF accepts the request, the event is not audited, and no warning message is issued.

WARNING

specifies that when a user requests creation of, or access to, a data set that is not RACF-protected, RACF is to allow the request but issue warning messages to the user and the security administrator.

NOPROTECTALL

specifies that a user can create or access a data set that is not protected by a profile.

NOPROTECTALL is in effect when RACF is using a newly-initialized database.

RACLIST | NORACLIST

RACLIST(*class-name* ...)

activates the sharing of in-storage profiles, both generic and discrete, for the classes specified. Also see GENLIST operand.

Activate this function to improve the performance of resource access checking for a general resource class. With the profiles for the class in storage, RACF requires database I/O when making an access decision.

A valid *class-name* is any member class for which RACLIST=ALLOWED is specified in the class descriptor table. Grouping classes are not valid, except for RACFVARS and NODES. If *class-name* is valid, not only the specified *class-name*, but all classes that share the same POSIT are processed. If some classes sharing the same POSIT are RACLIST=DISALLOWED, those classes are skipped.

If REFRESH is also specified, member classes for which RACLIST=DISALLOWED was specified are also valid because the SETROPTS RACLIST(*class-name*) REFRESH command refreshes classes that were RACLISTed by RACROUTE REQUEST=LIST,GLOBAL=YES or SETROPTS RACLIST. Likewise, classes for which SETROPTS GENLIST was specified are also valid.

You cannot SETROPTS RACLIST and SETROPTS GENLIST for the same general resource class.

If the following classes supplied by IBM are active, you *must* issue a SETROPTS RACLIST command:

APPCSERV	DEVICES	PROPCNTL	SECLABEL	SYSMVIEW
APPCTP	FIELD	PSFMPL	SERVAUTH	UNIXPRIV
CSFKEYS	NODES	PTKTDATA	STARTED	VTAMAPPL
CSFSERV	OPERCMDS	RACFVARS		

In-storage profiles for the following classes supplied by IBM can be optionally shared by using SETROPTS RACLIST. We recommend that you SETROPTS RACLIST the classes marked with a * if you define profiles in those classes and activate the classes:

ACCTNUM *	DBNFORM	JESINPUT	PERFGRP *	TERMINAL
ALCSAUTH	DCEUIDS	JESJOBS	PTKTVAL	TMEADMIN
APPCPORT	DIGTCERT *	JESSPOOL	RRSFDATA *	TSOAUTH *
APPCSI	DIGTRING	KEYSMSTR	SDSF	TSOPROC *

APPL	DLFCCLASS	LFSCCLASS	SERVER	VMATCH
CBIND	DSNR	LOGSTRM	SMESAGE	VMCMD
CONSOLE	FACILITY *	MGMTCLAS	SOMDOBJ	VMNODE
CPSMOBJ	FCICSFCT	MQCMDS	STORCLAS	VMSEGMT
CPSMXMP	INFOMAN	MQCONN	SUBSYSNM	WRITER
DASDVOL	JAVA	NETCMDS	SURROGAT	

If you have, or are considering, authorizing a large number of users for a resource in a class that can be processed to an in-storage profile using the SETROPTS RACLIST command, you must consider the number of entries in the access list, because RACLIST processing merges profiles and the access lists of each profile. The combined number of access-list entries may cause the profile to become too large to be processed, and RACLIST processing may fail. See *z/OS Security Server RACF Security Administrator's Guide* for more information about limiting the size of access lists and profile sizes.

Notes:

1. When you activate RACLIST processing for a class, RACF copies both discrete and generic profiles for that class into a data space.

When the RACGLIST class is active and class-name profiles have been specified in the RACGLIST class, SETROPTS RACLIST(*class-name*) stores the RACLISTed results from the data space in the RACGLIST classname_nnnnn profiles on the RACF database, enabling all systems sharing the database to access the same level of profile information.

For example if you issue the commands:

```
SETR CLASSACT(RACGLIST)
RDEFINE RACGLIST TERMINAL
```

Then either when you issue:

```
SETROPTS RACLIST(TERMINAL)
```

or at the next IPL, if the TERMINAL class was RACLISTed before the RACGLIST class was activated, RACF creates RACGLIST TERMINAL_00001, RACGLIST TERMINAL_00002, and so on, to hold the results of the SETROPTS RACLIST processing.

The profiles are available to all authorized users, thereby eliminating the need for RACF to retrieve a profile each time a user requests access to a resource protected by that profile. Thus, when you activate this function, you reduce processing overhead.

The SETROPTS RACLIST(*class-name*) command overrides a RACROUTE REQUEST=LIST,GLOBAL=YES request for the same class. The data space and RACGLIST classname_nnnnn profiles, if any, are refreshed by the SETROPTS RACLIST. SETROPTS LIST output will list the class in the "SETR RACLIST CLASSES =" line rather than the "GLOBAL=YES RACLIST ONLY =" line.

2. If you specify RACLIST with REFRESH, RACF rebuilds the discrete and generic profiles for the class and places them in the new data space. If

SETROPTS

the RACGLIST class is active and contains a profile for *class-name*, the *classname_nnnnn* profiles for the class are also rebuilt, or are created if they had not been built previously.

SETROPTS RACLIST(*class-name*) REFRESH can also be used to refresh classes RACLISTed by RACROUTE REQUEST=LIST,GLOBAL=YES, as well as classes that are RACLISTed. It refreshes the class, but has no effect on SETROPTS LIST output. If the class was processed using SETROPTS RACLIST solely by RACROUTE REQUEST=LIST, ENVIR=CREATE, GLOBAL=YES, the class are listed in the "GLOBAL=YES RACLIST ONLY =" line. Regardless of whether the class was RACLISTed by GLOBAL=YES, if it was RACLISTed by SETROPTS RACLIST (*classname*) then the class is listed only in the "SETR RACLIST CLASSES =" line.

SETROPTS RACLIST(*classname*) REFRESH can also be issued to create the RACGLIST profiles for the class, even if the class were not RACLISTed by either RACROUTE REQUEST=LIST,GLOBAL=YES or by SETROPTS RACLIST. Then the first RACROUTE REQUEST=LIST,GLOBAL=YES uses the RACLIST profiles to build the RACLIST data space, rather than accessing the database for each individual discrete and generic profile.

While the rebuild is in progress, RACF continues to use the old in-storage profiles for authorization requests until the new ones are created. When all systems have completed rebuilding the local data spaces, the coordinator signals the members of the data sharing group to discard the old ones, and to begin using the new one.

3. When RACF is enabled for sysplex communication, RACF propagates a SETROPTS RACLIST(*class-name*) or SETROPTS RACLIST(*class-name*) REFRESH command issued from any one system (coordinator) to other systems in the data sharing group (peers) if the command is successful on the system on which it was entered. If the RACGLIST *classname_profiles* were built for the class, peer members of the sysplex use the results to build the RACLIST data space on their system, but do not rebuild the RACGLIST profiles.

If a refresh is being done, RACF continues to use the old in-storage profiles for authorization requests until the new ones are created. When all systems have completed rebuilding the local data spaces, the coordinator signals the members of the data sharing group to discard the old ones, and to begin using the new one.

If RACF is not enabled for sysplex communication, you must issue the SETROPTS RACLIST(*class-name*) command and the SETROPTS RACLIST(*class-name*) REFRESH command on each system sharing the database.

NORACLIST(*class-name* ...)

deactivates the sharing of in-storage profiles, both generic and discrete, for the classes specified. Also see the NOGENLIST operand. For a list of such classes, see the description of the class descriptor table in *z/OS Security Server RACF Macros and Interfaces*.

When you specify NORACLIST, RACF deletes the data space containing the generic and discrete profiles for the specified classes. The data space might have been created by specifying the class with either a SETROPTS RACLIST command or a RACROUTE REQUEST=LIST,GLOBAL=YES request. In the latter case, all applications that issued a RACROUTE

SETROPTS

REQUEST=LIST,ENVIR=CREATE,GLOBAL=YES for the class should issue a RACROUTE REQUEST=LIST,ENVIR=DELETE before a SETROPTS NORACLIST is issued that processes the class. The SETROPTS NORACLIST should be used to delete the data space only after all applications have relinquished their access to it.

For both the SETROPTS RACLIST and RACROUTE REQUEST=LIST,GLOBAL=YES cases, if RACGLIST classname_nnnnn profiles exist for the class, they are deleted. Even if the class-name was not RACLISTed, SETROPTS NORACLIST can be used to delete these profiles. In all cases, the RACGLIST classname profile remains.

A valid *class-name* is any member class in the class descriptor table. Grouping classes are not valid, except for RACFVARS and NODES. If *class-name* is valid, not only the specified class but all classes that share the same POSIT are processed. Because SETROPTS NORACLIST, like SETROPTS RACLIST REFRESH, operates on classes that are RACLISTed by RACROUTE REQUEST=LIST,GLOBAL=YES, or SETROPTS RACLIST, member classes that are marked RACLIST DISALLOWED in the class descriptor table are now valid classes for the command. Both these conditions are still invalid for SETROPTS RACLIST.

When RACF is enabled for sysplex communication, RACF propagates the SETROPTS NORACLIST command to other systems in the data sharing group, if the command was successful on the system in which it was entered. If RACF is not enabled for sysplex communication, you must issue the SETROPTS NORACLIST command on each system sharing the database.

NORACLIST is in effect for all classes defined in the class descriptor table when RACF is using a newly-initialized database.

REALDSN | NOREALDSN

REALDSN

specifies that RACF is to record, in any SMF log records and operator messages, the real data set name (not the naming-conventions name) used on the data set commands and during resource access checking and resource definition.

NOREALDSN

specifies that RACF is to record, in any SMF log records and operator messages, the data set names modified according to RACF naming conventions.

NOREALDSN is in effect when RACF is using a newly-initialized database.

REFRESH

refreshes the in-storage generic profiles when specified with GENERIC, GLOBAL or RACLIST, or the in-storage program control tables when specified with WHEN(PROGRAM).

RETPD(*nnnnn*)

specifies the default RACF security retention period for tape data sets, where *nnnnn* is a 1-5 digit number in the range of 0 through 65533 or 99999 to indicate a data set that never expires. The security retention period is the number of days that RACF protection is to remain in effect for a tape data set; RACF stores the value in the tape data set profile.

If you specify RETPD, you must also specify TAPEDSN to activate tape data set protection. If you omit TAPEDSN, RACF records the value you specify for

SETROPTS

security retention period in the list of RACF options. However, without tape data set protection activated, this value is meaningless.

If you specify RETPD and TAPEDSN, the value you specify for security retention period is the default for your installation; RACF places the value in each tape data set profile unless the user specifies one of the following:

- An EXPDT in the JCL other than the current date
- An RETPD other than 0 on the ADDSD command.

If you specify TAPEDSN and do not specify RETPD, RACF uses a value of 0 for the default security retention period.

RVARYPW([SWITCH(*switch-pw*)] [STATUS(*status-pw*)])

specifies the passwords that the operator is to use to respond to requests to approve RVARY command processing, where *switch-pw* is the response to a request to switch RACF databases or change the operating mode of RACF, and *status-pw* is the response to a request to change RACF or database status from ACTIVE to INACTIVE or from INACTIVE to ACTIVE. You can specify different passwords for each response. Note that NO is not a valid password for either SWITCH or STATUS.

When RACF is using a newly-initialized database, the switch password and the status password are both set to YES.

SAUDIT | NOSAUDIT

specifies whether RACF is to log RACF commands issued by users with the SPECIAL or group-SPECIAL attribute. You must have the AUDITOR attribute to specify these operands.

SAUDIT

specifies that RACF is to log RACF commands (except LISTDSD, LISTGRP, LISTUSER, RLIST, and SEARCH) issued by users either had the SPECIAL attribute or who gained authority to issue the command through the group-SPECIAL attribute.

SAUDIT is in effect when RACF is using a newly initialized database.

NOSAUDIT

specifies that RACF is not to log the commands issued by users with the SPECIAL or group-SPECIAL attribute.

SECLABELAUDIT | NOSECLABELAUDIT

You must have the AUDITOR attribute to specify these options.

SECLABELAUDIT

Specifies that the SECLABEL profile's auditing options are to be used in addition to the auditing options specified for the resource profile. This additional auditing occurs whenever an attempt is made to access a resource protected by a profile that has a security label specified.

The SECLABEL profile requires SETROPTS RACLIST processing. If SECLABEL profile audit options are not specified, SECLABEL auditing is not done.

For more information, refer to *z/OS Security Server RACF Auditor's Guide*.

NOSECLABELAUDIT

disables auditing by SECLABEL.

NOSECLABELAUDIT is in effect when RACF is using a newly-initialized database.

SECLABELCONTROL | NOSECLABELCONTROL

SECLABELCONTROL

limits the users who can specify the SECLABEL operand on RACF commands. Those allowed to specify the operand are:

- Users with the SPECIAL attribute can specify the SECLABEL operand on any RACF command.
- Users with the group-SPECIAL attribute can specify the SECLABEL on the ADDUSER and ALTUSER commands when adding a user to a group within their scope of control (provided the group-SPECIAL is permitted to the SECLABEL).

NOSECLABELCONTROL

allows any user to change the SECLABEL field in a profile, as long as the user has at least READ access authority to the associated SECLABEL profile.

NOSECLABELCONTROL is in effect when RACF is using a newly-initialized database.

SECLEVELAUDIT | NOSECLEVELAUDIT

You must have the AUDITOR attribute to specify these operands.

SECLEVELAUDIT (*security-level*)

activates auditing of access attempts to all RACF-protected resources based on the specified installation-defined security level. RACF audits all access attempts for the specified security level and higher.

You can specify only a security level name defined by your installation as a SECLEVEL profile in the SECDATA class. (For information on defining security levels, see the description of the RDEFINE and RALTER commands.)

NOSECLEVELAUDIT

deactivates auditing of access attempts to RACF-protected resources based on a security level.

NOSECLEVELAUDIT is in effect when RACF is using a newly-initialized database.

SESSIONINTERVAL | NOSESSIONINTERVAL**SESSIONINTERVAL**(*n*)

sets the maximum value that can be specified by RDEFINE or RALTER for session key intervals. This value, *n*, must be from 1 to 32767 (inclusive).

The SESSIONINTERVAL value after RACF data set initialization is 30. This value is used for:

1. A default if SESSION is specified without INTERVAL on RDEFINE when defining an APPCLU class profile.
2. An upper limit if INTERVAL is specified on RDEFINE or RALTER for APPCLU class profiles.

NOSESSIONINTERVAL

disables the global limit on the number of days before a session key expires. The internal value is set to zero.

STATISTICS | NOSTATISTICS

use these operands to cause RACF to record or not record statistical information for the specified class name. The valid class names are DATASET and those classes defined in the class descriptor table. For a list of the general

SETROPTS

resource classes defined in the class descriptor table supplied by IBM, see Appendix B, “Description of RACF Classes”.

Note: If you activate or deactivate statistics processing for a class, all other classes in the class descriptor table with the same POSIT number are also be activated or deactivated. If, for instance, you activate statistics processing for the TIMS class, statistics processing is activated for classes AIMS and GIMS because they share POSIT number 5. For more information, see the description of the ICHERCDE macro in *z/OS Security Server RACF Macros and Interfaces*.

STATISTICS(*class-name* ... | *)

specifies that RACF is to record statistical information for *class-name*.

If you specify an asterisk (*), you activate the recording of statistical information for the DATASET class and all classes defined in the class descriptor table.

When RACF is using a newly-initialized database, the recording of class statistics is turned off. Because statistics recording has an impact on system performance, it is recommended that you do not activate this option for any class until your installation evaluates the need to use it versus the potential performance impact. For more information, see *z/OS Security Server RACF System Programmer's Guide*.

NOSTATISTICS(*class-name* ... | *)

specifies the names of the classes to be deleted from those previously defined to have statistical information recorded.

If you specify an asterisk (*), you deactivate the recording of statistical information for the DATASET class and all classes defined in the class descriptor table.

TAPEDSN | NOTAPEDSN

TAPEDSN

activates tape data set protection. When tape data set protection is in effect, RACF can protect individual tape data sets as well as tape volumes.

If you activate tape data set protection, you should also activate the TAPEVOL class. If you do not also activate TAPEVOL, RACF does not check the retention period before it deletes a tape data set, and you must provide your own protection for tape data sets that reside on a volume that contains more than one data set.

Before you activate tape data set protection, see *z/OS Security Server RACF Security Administrator's Guide* for a complete description of the relationship between TAPEDSN and activating the TAPEVOL class.

NOTAPEDSN

deactivates tape data set protection. When NOTAPEDSN is in effect, RACF cannot protect individual tape data sets, though it can protect tape volumes.

NOTAPEDSN is in effect when RACF is using a newly-initialized database.

TERMINAL(READ | NONE)

is used to set the universal access authority (UACC) associated with undefined terminals. If you specify TERMINAL but do not specify READ or NONE, the system prompts you for a value.

WHEN | NOWHEN

WHEN(PROGRAM)

activates RACF program control, which includes both access control to load modules and program access to data sets.

To set up access control to load modules, you must identify your controlled programs by creating a profile for each in the PROGRAM class. To set up program access to data sets, you must add a conditional access list to the profile of each program-accessed data set. Then, when program control is active, RACF ensures that each controlled load module is executed only by callers with the defined authority. RACF also ensures that each program-accessed data set is opened only by users who are listed in the conditional access list with the proper authority and who are executing the program specified in the conditional access list entry.

When RACF is enabled for sysplex communication, the SETROPTS WHEN(PROGRAM) command and the SETROPTS WHEN(PROGRAM) REFRESH command are propagated to other members of the data sharing group if the command was successful on the system on which it was entered. When RACF is not enabled for sysplex communication, you must issue the SETROPTS WHEN(PROGRAM) command and the SETROPTS WHEN(PROGRAM) REFRESH command on each system sharing the database.

For more information about program control, see *z/OS Security Server RACF Security Administrator's Guide*.

Note: The PROGRAM class does not have to be active.

NOWHEN(PROGRAM)

specifies that RACF program control is not to be active.

NOWHEN(PROGRAM) is in effect when RACF is using a newly-initialized database.

Examples

Table 71. SETROPTS Examples

Example 1

Operation User FRG34 wants to establish logging options that causes RACF to log all activity in the USER and GROUP classes, log the activities of users with the SPECIAL and group-SPECIAL attributes, log all accesses allowed only because the user has the OPERATIONS or group-OPERATIONS attribute, log all command violations, and audit all attempts to access RACF-protected resources based on the installation-defined security level "SECRET".

Known User FRG34 has the AUDITOR attribute. SECRET is defined as a SECLEVEL profile in the SECDATA class.

User FRG34 wants to issue this command as a RACF TSO command.

Command SETROPTS AUDIT(USER GROUP) OPERAUDIT SECLEVELAUDIT(SECRET)

Defaults SAUDIT CMDVIOL

SETROPTS

Table 71. SETROPTS Examples (continued)

Example 2	<i>Operation</i>	User RVU03 wants to establish a set of syntax rules for passwords that obey the following rules: <ul style="list-style-type: none">• The minimum password length is 4 characters• Four character passwords must have at least one numeric and one alphabetic character• Five character passwords must contain at least one numeric character or be completely alphabetic• Passwords of 6 or more characters consist of any combination of alphabetic and numeric characters.
	<i>Known</i>	User RVU03 has the SPECIAL attribute.
		User RVU03 wants to issue this command as a RACF TSO command.
	<i>Command</i>	SETROPTS PASSWORD(RULE1(LENGTH(4:5) ALPHANUM(1:5)) RULE2(LENGTH(5) ALPHA(1:5)) RULE3(LENGTH(6:8) ALPHANUM(1:8)) RULE4(LENGTH(6:8) NUMERIC(1:8)) RULE5(LENGTH(6:8) ALPHA(1:8)))
Example 3	<i>Defaults</i>	None
	<i>Operation</i>	User ADM1 wants to display the RACF options currently in effect. MVS and VM systems share the RACF database.
	<i>Known</i>	User ADM1 has the SPECIAL and AUDITOR attributes.
		User ADM1 wants to issue this command as a RACF TSO command.
Example 4	<i>Command</i>	SETROPTS LIST
	<i>Defaults</i>	None
	<i>Output</i>	See Figure 62 on page 474
	<i>Operation</i>	User RVU02 wants to establish system-wide options for an installation. The installation requires tape data set protection and tape volume protection, and the maximum password interval is to be 60 days. The default RACF security retention period for tape data sets is to be 360 days.
Example 5	<i>Known</i>	User RVU02 has the SPECIAL attribute.
		User RVU02 wants to issue this command as a RACF TSO command.
	<i>Command</i>	SETROPTS PASSWORD(INTERVAL(60)) CLASSACT(TAPEVOL) TAPEDSN RETPD(360)
	<i>Defaults</i>	None
Example 6	<i>Operation</i>	User ADM1 wants to enable the generic profile checking facility for the DATASET class.
	<i>Known</i>	User ADM1 has the SPECIAL attribute.
		User ADM1 wants to issue this command as a RACF TSO command.
	<i>Command</i>	SETROPTS GENERIC(DATASET)
Example 7	<i>Defaults</i>	None
	<i>Operation</i>	User ADM1 wants to activate global access checking for the DATASET class.
	<i>Known</i>	User ADM1 has the SPECIAL attribute.
		User ADM1 wants to issue this command as a RACF TSO command.
Example 8	<i>Command</i>	SETROPTS GLOBAL(DATASET)
	<i>Defaults</i>	None
	<i>Operation</i>	User ADM1 wants to activate erase-on-scratch processing for all resources with a security level of CONFIDENTIAL or higher and set the SWITCH and STATUS passwords for the RVARY command.
	<i>Known</i>	User ADM1 has the SPECIAL attribute. The CONFIDENTIAL security level name is known to RACF.
Example 9		User ADM1 wants to issue this command as a RACF TSO command.
	<i>Command</i>	SETROPTS ERASE(SECLEVEL(CONFIDENTIAL)) RVARYPW(SWITCH(LINUS) STATUS(LUCY))
	<i>Defaults</i>	None

Table 71. SETROPTS Examples (continued)

Example 8	<i>Operation</i>	The RACF system administrator wants to activate installation defaults for the primary and secondary national languages. The primary language is Japanese and the secondary language is Canadian French.
	<i>Known</i>	The system administrator has the SPECIAL attribute. RACF is running under MVS/ESA SP™ Release 4.2 but the message service is not active. The 3 character language code for Japanese is JPN. The language code for Canadian French is FRC.
		The system administrator wants to issue this command as a RACF TSO command.
	<i>Command</i>	SETROPTS LANGUAGE(PRIMARY(JPN) SECONDARY(FRC))
	<i>Defaults</i>	None

SETROPTS

```
SETROPTS LIST1
ATTRIBUTES = INITSTATS NOWHEN(PROGRAM) TERMINAL(READ) SAUDIT CMDVIOL NOOPERAUDIT
STATISTICS = DATASET DASDVOL TAPEVOL TERMINAL APPL TMS GIMS AIMS TCICSTRN
              GCICSTRN PCICSPSB QCICSPSB
AUDIT CLASSES = DATASET USER GROUP DASDVOL TAPEVOL TERMINAL APPL TMS
                 GIMS AIMS TCICSTRN GCICSTRN PCICSPSB QCICSPSB
ACTIVE CLASSES = DATASET USER GROUP DASDVOL TAPEVOL TERMINAL APPL TMS
                 GIMS AIMS TCICSTRN GCICSTRN PCICSPSB QCICSPSB FCICSFCT
                 HCICSFCT JCICSJCT KCICSJCT DCICSDCT ECICSDCT SCICSTST
                 UCICSTST MCICSPPT NCICSPPT ACICSPCT BCICSPCT CCICSCMD
                 VCICSCMD GMBR GLOBAL VMRDR VMMDISK RACGLIST
GENERIC PROFILE CLASSES = DATASET DASDVOL TAPEVOL TERMINAL APPL TMS
                          AIMS TCICSTRN PCICSPSB FCICSFCT JCICSJCT
                          DCICSDCT SCICSTST MCICSPPT ACICSPCT CCICSCMD
                          GMBR VMMDISK VMRDR VMCMD VMNODE VMBATCH
GENERIC COMMAND CLASSES = DATASET DASDVOL TAPEVOL TERMINAL APPL TMS
                          AIMS TCICSTRN PCICSPSB FCICSFCT JCICSJCT
                          DCICSDCT SCICSTST MCICSPPT ACICSPCT CCICSCMD
                          GMBR VMMDISK VMRDR VMCMD VMNODE VMBATCH
GENLIST CLASSES = NONE
GLOBAL CHECKING CLASSES = VMMDISK
SETR RACLIST CLASSES = DASDVOL ACCTNUM
GLOBAL=YES RACLIST ONLY = TCICSTRN JCICSJCT
LOGOPTIONS "ALWAYS" CLASSES = DASDVOL GDASDVOL SECLABEL
LOGOPTIONS "NEVER" CLASSES = VXMBR VMXEVENT FACILITY
LOGOPTIONS "SUCCESSES" CLASSES = RVARSMBR RACFVARS APPCLU
LOGOPTIONS "FAILURES" CLASSES = PMBR PROGRAM DATASET PROPCNTL
LOGOPTIONS "DEFAULT" CLASSES = TAPEVOL TERMINAL GTERMINL
AUTOMATIC DATASET PROTECTION IS IN EFFECT
ENHANCED GENERIC NAMING IS IN EFFECT
REAL DATA SET NAMES OPTION IS ACTIVE
JES-BATCHALLRACF OPTION IS INACTIVE
JES-XBMALLRACF OPTION IS INACTIVE
JES-EARLYVERIFY OPTION IS INACTIVE
PROTECT-ALL OPTION IS NOT IN EFFECT
TAPE DATA SET PROTECTION IS ACTIVE
SECURITY RETENTION PERIOD IN EFFECT IS   365 DAYS
ERASE-ON-SCRATCH IS INACTIVE
SINGLE LEVEL NAME PREFIX IS RDSRFX
LIST OF GROUPS ACCESS CHECKING IS ACTIVE.
INACTIVE USERIDS ARE NOT BEING AUTOMATICALLY REVOKED.
DATA SET MODELLING NOT BEING DONE FOR GDGS.
USER DATA SET MODELLING IS BEING DONE.
GROUP DATA SET MODELLING IS BEING DONE.
```

¹ The second line of this display, "ATTRIBUTES =", refers to global RACF attributes in effect. These attributes can be set only with the SETROPTS command. They are different from, and should not be confused with, the RACF user attributes.

Figure 62. Output for Example 3: SETROPTS LIST (Part 1 of 2)

```

PASSWORD PROCESSING OPTIONS:
  PASSWORD CHANGE INTERVAL IS 254 DAYS.
  NO PASSWORD HISTORY BEING MAINTAINED.
  USERIDS NOT BEING AUTOMATICALLY REVOKED.
  NO PASSWORD EXPIRATION WARNING MESSAGES WILL BE ISSUED.
INSTALLATION PASSWORD SYNTAX RULES:
  RULE 1  LENGTH(4:5)  LLLLL
  RULE 2  LENGTH(5)    AAAAA
  RULE 3  LENGTH(6:8)  LLLLLLL
  RULE 4  LENGTH(6:8)  NNNNNNN
  RULE 5  LENGTH(6:8)  AAAAAAA
LEGEND:
  A-ALPHA C-CONSONANT L-ALPHANUM N-NUMERIC V-VOWEL W-NOVOWEL *-ANYTHING
DEFAULT RVAR PASSWORD IS IN EFFECT FOR THE SWITCH FUNCTION.
DEFAULT RVAR PASSWORD IS IN EFFECT FOR THE STATUS FUNCTION.
SECLEVELAUDIT IS INACTIVE
SECLABEL AUDIT IS IN EFFECT
SECLABEL CONTROL IS IN EFFECT
GENERIC OWNER ONLY IS IN EFFECT
COMPATIBILITY MODE IS IN EFFECT
MULTI-LEVEL QUIET IS IN EFFECT
MULTI-LEVEL STABLE IS IN EFFECT
MULTI-LEVEL SECURE IS IN EFFECT. CURRENT OPTIONS:
  "MLS WARNING" OPTION IS IN EFFECT
MULTI-LEVEL ACTIVE IS IN EFFECT. CURRENT OPTIONS:
  "MLACTIVE FAIL" OPTION IS IN EFFECT
CATALOGUED DATA SETS ONLY, IS IN EFFECT. CURRENT OPTIONS:
  "CATDSNS WARNING" OPTION IS IN EFFECT
USER-ID FOR JES NJEUSERID IS : ????????
USER-ID FOR JES UNDEFINEDUSER IS : ++++++++
PARTNER LU-VERIFICATION SESSIONKEY INTERVAL MAXIMUM/DEFAULT IS    30 days
APPLAUDIT IS IN EFFECT
PRIMARY LANGUAGE DEFAULT : ENU / AMERICAN
SECONDARY LANGUAGE DEFAULT : ENU / AMERICAN
ADDCREATOR IS IN EFFECT
KERBLVL = 0
EIM REGISTRY = DCEIMGUI-RACF

```

Note: The language name (in this example, AMERICAN) only appears if the MVS message service is active.

Figure 62. Output for Example 3: SETROPTS LIST (Part 2 of 2)

SIGNOFF (Sign Off Sessions)

Background

Persistent verification allows users to sign on to a partner LU (logical unit) and have their authority persist. In other words, once a user has signed on, a password is not required for subsequent sign-on attempts.

APPC/MVS invokes RACF to create and maintain a list called the signed-on-from list. If persistent verification is being used, the signed-on-from list consists of the users currently signed on with Persistent Verification authority.

Purpose

The RACF SIGNOFF operator command removes user entries from the signed-on-from list. Entries in the signed-on-from list are selected by the SIGNOFF command using the following information:

- User ID
- Group
- APPL (the local LU name)
- POE (the partner LU name from which the user is signed on)

The SIGNOFF command has operands which correspond to the items above. You can use these operands to select which user entries to remove from the signed-on-from list.

To determine which user entries are signed off by issuing a particular SIGNOFF command, issue a DISPLAY command with corresponding selection criteria.

Issuing Options

The following table identifies the eligible options for issuing the SIGNOFF command:

Table 72. How the SIGNOFF Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
No	Yes	No	No	Yes

Note: For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

Use the DISPLAY operator command to view the signed-on-from list.

Authorization Required

When issuing the SIGNOFF command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the SIGNOFF command is:

<pre> subsystem-prefix SIGNOFF APPL(local-luname *) POE(partner-luname *) USER(userid-name *) [GROUP(group-name *)] [SECLABEL(security-label *)] </pre>
--

Note: For additional information on issuing this command as a RACF operator command, refer to “Rules for entering RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

The operands listed below allow the operator to specify the user entries to be signed off. The **APPL**, **POE** and **USER** operands are required to uniquely identify a user entry to be signed off. The **GROUP** operand is optional and defaults to a *group-name* consisting of blanks.

APPL(*local-luname* | *)

This is a required operand. The *local-luname* is a 1-8 character name of the local LU to be searched for. An asterisk can occupy the last position of the *local-luname* in order to provide a partial generic selection capability. A character string consisting of a single asterisk is permitted as a full generic that matches any APPL name in the signed-on-from list.

POE(*partner-luname* | *)

This is a required operand. The *partner-luname* is a 1-17 character name of the partner LU to be searched for. A *partner-luname* consisting of a single asterisk is permitted as a full generic that matches any POE name in the signed-on-from list.

USER(*userid-name* | *)

This is a required operand. The *userid-name* is a 1-8 character specification which represents the RACF user ID to be searched for. A character string consisting of a single asterisk is permitted as a full generic that matches any user ID in the signed-on-from list.

GROUP(*group-name* | *)

This is an optional operand. The *group-name* is a 1-8 character name which represents the RACF group to be searched for. A character string consisting of

SIGNOFF

a single asterisk is also permitted as a full generic which matches any *group-name* in the signed-on-from list. If this operand is not specified, the default value is a *group-name* consisting of blanks.

Note that entries in the signed-on-from list might not always be added to that list with a *group-name* value. Such entries have *group-names* consisting of blanks.

SECLABEL(*security-label* | *)

This is an optional operand. The *security-label* is a 1-8 character name that represents the RACF security label to be searched for. This operand is currently ignored.

Examples

Table 73. SIGNOFF Examples

Example 1	<p><i>Operation</i> Sign off a user from a local/partner LU pair.</p> <p><i>Known</i> The local LU is locallu, the partner LU is prtnrlu5, and the <i>userid-name</i> is jim. The RACF subsystem prefix is @.</p> <p><i>Command</i> @signoff user(jim),appl(locallu),poe(prtnrlu5)</p> <p><i>Defaults</i> A <i>group-name</i> consisting of blank characters.</p>
Example 2	<p><i>Operation</i> Sign off all of the users from a local/partner LU pair.</p> <p><i>Known</i> The local LU is locallu, the partner LU is prtnrlu5, and the RACF subsystem prefix is @.</p> <p><i>Command</i> @signoff appl(locallu),poe(prtnrlu5),user(*)</p> <p><i>Defaults</i> A <i>group-name</i> consisting of blank characters.</p>
Example 3	<p><i>Operation</i> Sign off a user from all the local/partner LU pairs to which that user is signed on.</p> <p><i>Known</i> The <i>userid-name</i> is Kurt, and the RACF subsystem prefix is @.</p> <p><i>Command</i> @signoff appl(*),poe(*),group(*),user(jim)</p> <p><i>Defaults</i> None</p>
Example 4	<p><i>Operation</i> Sign off all users from all the partner LUs of a particular local LU.</p> <p><i>Known</i> The local LU is locallu, the RACF subsystem prefix is @.</p> <p><i>Command</i> @signoff appl(locallu),poe(*),user(*),group(*)</p> <p><i>Defaults</i> None</p>
Example 5	<p><i>Operation</i> Sign off all users of a particular group from a particular local LU.</p> <p><i>Known</i> The local LU is locallu, the group is grp1, and the RACF subsystem prefix is @.</p> <p><i>Command</i> @signoff appl(locallu),poe(*),user(*),group(grp1)</p> <p><i>Defaults</i> None</p>

STOP (Shutdown RRSF)

Purpose

Use the STOP command to allow the MVS operator to stop the RACF subsystem address space if restarting a subtask is not sufficient to recover from a failure. This command shuts down the RACF subsystem address space and prevents the loss of any outstanding requests that are waiting for completion.

The STOP command can also be used to stop the RACF subsystem address space before an IPL. In releases before RACF 2.2, it was not necessary to stop the RACF subsystem address space before IPLing. On RACF 2.2 or later, however, if you are directing work to the RACF subsystem, you should stop the RACF subsystem address space before IPLing to prevent the loss of outstanding requests. Note, though, that if you are using automatic direction or password synchronization you should not stop the address space except immediately before an IPL. If you stop it at other times, updates that are made to the local node might not be sent to the other nodes.

Note: All users or applications that update the RACF database should be completed before issuing the STOP command.

Issuing Options

The following table identifies the eligible options for issuing the STOP command:

Table 74. How the STOP Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
No	Yes	No	No	No

Note: For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

- To restart the RACF subsystem address space, use the MVS START command.
- To restart a function in the RACF subsystem address space, see “RESTART (Restart RRSF Functions)” on page 379. You must restart the RACF subsystem address space before restarting functions within the RACF subsystem address space.

Authorization Required

When issuing the STOP command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator's Guide* for further information.

Syntax

The complete syntax of the command is:

<i>subsystem-prefix</i> STOP

STOP

Note: For additional information on issuing this command as a RACF operator command, refer to “Rules for entering RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

The subsystem prefix is a required keyword for RACF operator commands.

Examples

Table 75. STOP Example

Example

Operation User AMT02 wants to shut down the RACF subsystem address space in an orderly manner without losing any remote RRSF requests.

Known The RACF subsystem prefix is @.

Command @STOP

Defaults None

TARGET (Define RRSF Nodes)

Purpose

Use the TARGET command to:

- List the controls and operational characteristics of the specified target RRSF nodes.
- Specify the name of the target RRSF node.
- Request an operational state for connection to the target RRSF node.
- Delete an RRSF node from the set of target nodes known to the local node.
- Specify a description of the target RRSF node.
- Purge the workspace data sets managed by RRSF in the RACF subsystem address space.
- Specify the protocol type for the transport mechanism to be used in communication between the two RRSF remote nodes.
- Specify a prefix for the workspace data sets allocated by and used by RRSF for each target node.
- Specify the characteristics of the workspace data sets associated with the node being defined to RRSF.

Note: You might find it useful to fill out the “Configuration Worksheet” or read the the chapter on RRSF in the *z/OS Security Server RACF System Programmer’s Guide* to help you determine the information you need to issue the TARGET command.

Issuing Options

The following table identifies the eligible options for issuing the TARGET command:

Table 76. How the TARGET Command Can be Issued

As a RACF TSO Command?	As a RACF Operator Command?	With Command Direction?	With Automatic Command Direction?	From the RACF Parameter Library?
No	Yes	No	No	Yes

Note: For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related Commands

To set RRSF operational controls, see “SET” on page 420.

Authorization Required

When issuing the TARGET command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator’s Guide* for further information.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the command is:

TARGET

subsystem-prefix TARGET

```
[ DELETE | DORMANT | OPERATIVE ]
[ DESCRIPTION('description') ]
[ LIST ]
[ LOCAL ]
[ MAIN ]
[ NODE(nodenamel*) ]
[ PREFIX(qualifier...) ]
[ PROTOCOL(APPCLUNAME(luname)
  [ TPNAME(profile-name) ]
  [ MODENAME(mode-name) ] ) ) ]
[ PURGE(INMSG | OUTMSG) ]
[ SYSNAME(sysname | *) ]
[ WDSQUAL(qualifier) ]
[ WORKSPACE( {
  [ STORCLASS(class-name) ]
  [ DATACLAS(class-name) ]
  [ MGMTCLAS(class-name) ]
  | [ VOLUME(volume-serial) ] }
  [ FILESIZE([nnnnnnnnnnl500]) ] ) ]
```

Note: For additional information on issuing this command as a RACF operator command, refer to “Rules for entering RACF operator commands” on page 21.

Parameters

subsystem-prefix

specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the 1-8 character installation-defined prefix for RACF or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

DELETE | DORMANT | OPERATIVE

DELETE

specifies that the target node is to be deleted from the set of known targets.

Subsequent attempts to perform operations requiring the existence of the deleted target fail and a message is issued.

When a node is deleted, any workspace data sets that are currently allocated are deallocated. If the workspace data sets are empty, they are also deleted. The local node can only be deleted after all other nodes are deleted.

The local system, the system represented by the target command whose SYSNAME matches the CVTSNAME of the system the command is to run on, cannot be deleted until all other TARGET definitions have been deleted.

The local MAIN system cannot be deleted until all of its remote target definitions have been deleted.

TARGET

The MAIN system of a remote multisystem node cannot be deleted until all other peer systems of that remote multisystem node are deleted.

The DELETE keyword cannot be specified for a node whose connection is operative. The connection must first be made dormant.

Notes:

1. The DELETE keyword cannot be specified with the DORMANT or the OPERATIVE keywords. The only keywords that can be issued with the DELETE keyword are NODE, LOCAL, PURGE and SYSNAME.
2. If the PURGE keyword is specified with the DELETE keyword, the PURGE keyword is processed first regardless of the order in which the keywords were specified.

DORMANT

specifies that the connection to the target is to be made inactive.

While a node is dormant, RRSF queues requests for the node and holds them on the OUTMSG queue until the connection to the node is made operative by an authorized operator issuing a TARGET OPERATIVE command.

Notes:

1. The DORMANT keyword cannot be specified with the DELETE or the OPERATIVE keyword.
2. If the PURGE keyword is specified with the DORMANT keyword, the DORMANT keyword is processed first regardless of the order in which the keywords were specified.
3. If either the remote node and/or the local node is a multisystem node, you must define a MAIN system to multisystem node before making the conversation dormant.

OPERATIVE

specifies that the connection to the target node is to be made active.

As a result, all new and previously held requests for the node are sent.

The OPERATIVE keyword can only activate the connection to the target node if sufficient information has been provided on the WORKSPACE and PROTOCOL keywords and the PREFIX keyword has been specified.

Notes:

1. The OPERATIVE keyword cannot be specified with the DELETE or the DORMANT keyword.
2. If either the remote node and/or the local node is a multisystem node, you must define a MAIN system to the multisystem node before making the conversation operative.
3. If the PURGE keyword is specified with the OPERATIVE keyword, the PURGE keyword is processed first regardless of the order in which the keywords were specified.
4. If a connection is already operative and the OPERATIVE keyword is specified, the connection to that node is refreshed. If the node is the local node and the APPC server has been registered, the server is re-registered. The APPC transaction program (TP program) is restarted for local and remote nodes.

DESCRIPTION('description')

specifies a comment used to describe this target node.

TARGET

The following apply to *description*:

- The maximum length of the description is 32 characters.
- The description can consist of any characters, and can be entered with or without single quotes.
- If parentheses, commas, blanks, or semicolons are part of the description, the character string must be enclosed in single quotes.
- If a single quote is part of the description, and the entire character string is enclosed in single quotes, two single quotes must be entered together for each single quote within the character string.
- If the first character of the description is a single quote, then the string must be entered within single quotes and two single quotes must be entered for the first character.
- If the description is entered in lowercase, it is changed to uppercase.

The description is displayed in the TARGET LIST output.

LIST

lists the current operational and protocol attributes of the target specified on the NODE keyword or all targets (*).

For a multisystem node, the LIST keyword can display information about the node and about the systems that make up the node.

The LIST keyword is the default on the TARGET command in the following situations:

- No other keyword is specified,
- Only the NODE(*) keyword is specified
- Only the NODE(*nodename*) keyword is specified,
- Only the NODE(*nodename*) and SYSNAME(*sysname*) keywords are specified,
- Only the NODE(*nodename*) and SYSNAME(*) keywords are specified, or
- Only the NODE(*) and SYSNAME(*) keywords are specified.

If the LIST keyword is specified in combination with any other TARGET command keyword, the information displayed provides the operational and protocol attributes as they exist after the processing of the other keywords.

If a request to list information specifies NODE(*nodename*), then a detailed list for that one target node is provided. If a request to list information specifies NODE(*), then a detailed list, sorted by target node name, is provided for all known nodes including systems on a multisystem node. If the NODE keyword is not specified, then a less detailed summary list is provided for all target nodes known to the node the command runs on, sorted by target node name.

The LIST keyword can display a summary or detailed list of information for an RRSF node. The type of information displayed, depends on the following combinations of NODE and SYSNAME:

NODE(*)

detailed list of every node and system

NODE(*nodename*) SYSNAME(*)

detailed list of every system in the multisystem node

NODE(nodename)

for a multisystem node, summary list of every system in the multisystem node

NODE(nodename)

for a single-system node, detailed list of the single-system node

NODE(nodename) SYSNAME(systemname)

detailed list of the particular system

NODE(*) SYSNAME(*)

detailed list of every node and system

LOCAL

identifies the node specified on the NODE keyword as the local node and all other keywords as local node specifications.

Only one local node can be defined. Once you have defined a node as the local node, you do not need to specify LOCAL on subsequent TARGET commands issued for that node or for systems being added to the local multisystem node. The local node must be defined before attempting to make any other nodes operative since the local SYSTEM name is needed to allocate the workspace data sets.

If you do not specify LOCAL on the TARGET command, the specified node is defined as a remote node.

MAIN

identifies the system named on the SYSNAME keyword as the main system in a multisystem RRSF node. You must issue a TARGET command identifying the main system for a multisystem node on each system in the multisystem node, and on each RRSF node that communicates with the multisystem node. You must designate the same system as the main system on the local node and on all nodes that communicate with it. You must identify the main system of a multisystem node before you make any systems in the multisystem node operative or dormant.

NODE(nodename*)

specifies the name of the RRSF node that is being defined or is already defined and is being listed or modified by the specified keywords.

All nodes that communicate with the local node and the local node itself must be defined with a TARGET command.

The specified node name must:

- Be 1-8 characters in length
- Begin with either A-Z, #, \$, or @
- Use only A-Z, 0-9, #(X'7B'), \$(X'5B'), or @(X'7C') as the second through eighth characters

The asterisk (*) can only be specified with the LIST operand. If other keywords are specified with NODE(*), the command fails with a n error message.

PREFIX(qualifier...)

specifies the high-level qualifiers that RRSF uses for the workspace data set names of the INMSG and OUTMSG queues for the RRSF node defined by the TARGET command. The WORKSPACE keyword specifies the attributes of the workspace data sets.

The maximum length of the prefix is 19 characters. Multiple qualifiers can be specified, separated by periods. For example,

TARGET

PREFIX(*qualifier1.qualifier2...qualifiern*) specifies multiple qualifiers that serve as the highest level qualifiers of the workspace data set names.

RRSF uses the PREFIX to allocate workspace data sets for the node specified by the TARGET commands. If the workspace data sets are already allocated, the prefix cannot be changed and an error message is issued. When selecting a prefix, ensure that the workspace data sets are protected by a RACF DATASET profile, and that the user ID associated with the RACF subsystem address space has authority to create and access them. It is recommended that all systems of the same multisystem node (whether LOCAL or REMOTE) have the same prefix.

PROTOCOL

Protocol information is required for communication with remote nodes. Do not specify the PROTOCOL keyword when the local node is to run in local mode. If you specify the PROTOCOL keyword, APPC is required for the local node to go OPERATIVE and unnecessary processing occurs.

APPC

specifies that APPC is used as the network transport mechanism with the specified operands.

LUNAME(*luname*)

LUNAME is a required operand of the APPC keyword for all nodes if a remote node is to be made operative through the TARGET command. *Luname* is the logical unit name which is associated with the target RRSF node being described. It can be a 1-8 character unqualified LU name or a 1-17 character qualified LU name in the format *netid.luname*, where *netid* and *luname* are each 1-8 characters.

No validity checking is performed for LU name. Further, if the same LU name is defined on multiple TARGET commands, the first occurrence is the one used. For example, if two TARGET commands are issued for two different RRSF node names, but with the same LU name, the first RRSF node is associated with LU name and an error message is issued indicating that a second usage of LU name has been detected.

With the multisystem node support, the LU name can be modified only if the node is in the initial or defined state.

The information needed for this operand is in the SYS1.PARMLIB member APPCPMxx of the target RRSF node.

When building the data set name for the workspace files, TARGET uses only the second qualifier of the LU name. If this qualifier is not unique within the group of DASD devices shared by the local system, you are required to use the WDSQUAL keyword to supply a unique qualifier for the workspace data set name. For remote nodes, TARGET uses WDSQUAL as the third qualifier of the data set name so that the data set name appears as follows:

```
prefix.local-luname.{remote-luname OR WDSQUAL-value}.{INMSG  
or OUTMSG}
```

The output of the TARGET LIST command contains the qualified LU name if it is available.

TPNAME(*profile-name*)

profile-name specifies the APPC transaction program (TP) profile. The TP profile to be used at the target can be specified as a character string, 1-64 characters in length.

The default is 'IRRRACF'.

Once the *profile-name* is specified for a node, it can only be changed if the node is first made dormant.

MODENAME(*mode-name*)

modename is the mode name which designates the network properties for the session to be allocated. The APPC mode (logmode) used can be specified as an alphanumeric string, 1-8 characters in length. If no *modename* is specified, RACF supplies a MODENAME default. However, the TARGET LIST output shows the MODENAME as <NOT SPECIFIED>. Please refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for further information.

Once the mode name is specified for a node, it can only be changed if the node is first made dormant.

The default is IRRMODE.

PURGE (INMSG | OUTMSG)

specifies that all entries for the specified INMSG or OUTMSG workspace data set for the specified target node are to be purged.

INMSG and OUTMSG workspace data sets can only be purged if the node is first made dormant.

Note: If the PURGE keyword is specified with the DELETE or OPERATIVE keyword, the PURGE keyword is processed first regardless of the order in which the keywords were specified.

If the PURGE keyword is specified with the DORMANT keyword, the DORMANT keyword is processed first regardless of the order in which the keywords were specified.

SYSNAME(*sysname* | *)

identifies which system on a multisystem node the command pertains to. If the SYSNAME keyword is specified, the NODE keyword must also be specified. The SYSNAME keyword is required on TARGET commands for multisystem nodes. If it is not specified, RACF assumes that the node is a single-system node. The SYSNAME keyword is not required if the LIST keyword is specified or defaulted to.

Sysname is 1-8 characters long, and can contain the characters A-Z, 0-9, \$, #, @. It should not have a numeric as the first character, because it is used as a data set qualifier for the local workspace data sets. *sysname* must match the value in the CVTSNAME field of the system that the TARGET command describes. The CVTSNAME is the SYSNAME specified in the IEASYSxx member of SYS1.PARMLIB.

You can specify an asterisk (*) on the SYSNAME keyword to represent all systems currently defined for the node name specified by the NODE keyword. The asterisk (*) is only allowed to be specified with the NODE, DORMANT, OPERATIVE, DELETE, PURGE, and LIST keywords. When specified with DORMANT, OPERATIVE, DELETE, or PURGE, the requested action is attempted for all systems defined for that multisystem node. You can specify an asterisk (*) on the SYSNAME keyword when the LIST keyword is specified, causing a list to be generated for each system in the multisystem node specified on the NODE keyword. If NODE(*) is specified, SYSNAME must be specified as SYSNAME(*) or omitted.

TARGET

The SYSNAME keyword allows you to use a common set of TARGET commands on all the systems in a multisystem node. When the TARGET command is for a local node, and the OPERATIVE or DORMANT keyword is specified, RACF compares the SYSNAME specified on the TARGET command with the CVTSNAME for the system the command is to run on. If they do not match, RACF does not process the OPERATIVE or DORMANT keyword. The same is true for defined systems whose SYSNAMEs do not match the CVTSNAME when SYSNAME(*) is specified with OPERATIVE or DORMANT for a local node. In addition, because a conversation should not exist between the systems of a multisystem node, RACF issues an informational message and places it in the SYSLOG. This message might help diagnose why an expected conversation was not established.

WDSQUAL(*qualifier*)

specifies the middle qualifier that RRSF uses for the workspace data set names of the INMSG and OUTMSG queues for the local RRSF node defined by the TARGET command. When specified on the TARGET command for a remote node, it specifies a replacement qualifier for the remote LU name as the third qualifier of the workspace data set names for that node. The WORKSPACE keyword specifies the attributes of the workspace data sets.

The maximum length of the workspace data set name qualifier is 8 characters, and the first character must be an alphabetic character.

RRSF uses the qualifier to allocate workspace data sets for the node specified by the TARGET commands. If the workspace data sets are already allocated, the qualifier cannot be changed and an error message is issued. When selecting a qualifier, ensure that the workspace data sets are protected by a RACF DATASET profile, and that the user ID associated with the RACF subsystem address space has authority to create and access them.

The naming convention for the workspace data sets created on a node as a result of a TARGET command for a remote node is: `prefix.local-luname.remote-luname-or-wdsqual.ds-identity`

Where:

prefix

Is a value you specify with the PREFIX keyword on the TARGET command.

local-luname

Is the LU name of the local node.

remote-luname-or-wdsqual

Is the LU name of the remote node if a workspace data set name qualifier is not specified on the TARGET command. If a workspace data set name qualifier is specified on the TARGET command for a remote node, then it is used instead of the LU name for that node.

ds-identity

Is either INMSG or OUTMSG.

The naming convention for the workspace data sets created on a node as a result of a TARGET LOCAL command is:

`prefix.sysname-or-wdsqual.ds-identity`

Where:

prefix

Is a value you specify with the PREFIX keyword on the TARGET command.

sysname-or-wdsqual

Is the system name if the workspace data set name qualifier is not specified on the TARGET command. The SYSNAME value must match the value in the CVTSNAME field for the system it identifies. If a workspace data set name qualifier is specified on the TARGET command, it is used instead of the SYSNAME value.

ds-identity

Is either INMSG or OUTMSG.

WORKSPACE

The WORKSPACE keyword, combined with its operands, specifies information to be used during the allocation of two work data sets. The data sets are used by RRSF to control information and messages associated with work being processed by the RACF subsystem address space for the specified node. There are two data sets for each target node. No workspace data sets are allocated for peer systems in the same multisystem node or for non-main members between two multisystem nodes. For multisystem nodes, it is recommended that all workspace data sets are allocated on shared resources so that all member systems of the multisystem node can access any workspace data set.

The two data sets are allocated by RRSF using the information provided on the WORKSPACE and PREFIX keywords to determine their fully qualified names, data set size and other data-management-related information. RRSF provides the low level qualifiers of the data set names as follows:

- **INMSG**

The INMSG data set is used by RRSF as a queue to keep track of requests that are associated with inbound work or work that is to be delivered.

- **OUTMSG**

The OUTMSG data set is used by RRSF as a queue to keep track of all directed requests.

The request remains in the data set until confirmation is received that the specified target node has received the request, and is then removed.

The naming convention for the workspace data sets created on a node as a result of a TARGET LOCAL command is:

- *prefix.sysname_or_wdsqual.ds_identity*

Where:

prefix

Is a value you specify with the PREFIX keyword on the TARGET command.

sysname_or_wdsqual

Is the system name if a workspace data set name qualifier is not specified on the TARGET command. The SYSNAME must match the value in the CVTSNAME field for the system it identifies. If a workspace data set name qualifier is specified on the TARGET command, then it is used instead of the SYSNAME.

ds_identity

Is either INMSG or OUTMSG.

The naming convention for the workspace data sets for remote connections is:

- *prefix.local-luname.remote-luname-or-wdsqual.ds-identity*

Where:

TARGET

prefix

Is a value you specify with the PREFIX keyword on the TARGET command.

local-luname

Is the LU name of the local node.

remote-luname-or-wdsqual

Is the LU name of the remote node if a workspace data set name qualifier is not specified on the TARGET command. If a workspace data set name qualifier is specified on the TARGET command for a remote node, then it is used instead of the LU name for that node.

ds_identity

Is either INMSG or OUTMSG.

You can specify either an SMS or non-SMS WORKSPACE. The STORCLASS, DATACLAS, and MGMTCLAS keywords apply to an SMS WORKSPACE. The VOLUME keyword applies to a non-SMS WORKSPACE. The FILESIZE keyword applies to both.

You cannot specify the VOLUME keyword with any of the SMS keywords (STORCLAS, MGMTCLAS, or DATACLAS).

STORCLAS(*class-name*)

specifies the SMS storage class name. This must be specified if SMS is being used for the workspace.

DATACLAS(*class-name*)

specifies the SMS data class name. This keyword applies if SMS is being used for the workspace.

MGMTCLAS(*class-name*)

specifies the SMS management class name. This keyword applies if SMS is being used for the workspace.

VOLUME(*volume-serial*)

specifies the volume serial number of the volume that contains the RRSF workspace data sets. This is primarily a configuration value that describes where the next new workspace data set is allocated.

The *volume-serial* specified must be a valid volume on the system where the TARGET command is issued.

FILESIZE(*nnnnnnnnnn*500)

specifies that space allocated for the workspace data sets should be sufficient to contain *nnnnnnnnnn* entries for each of the data sets.

The range of values that can be specified is from 1 to 2147483647.

FILESIZE(500) is the initial value.

Note: The value supplied for FILESIZE is used to supply a RECORDS parameter when RACF invokes Access Methods Services DEFINE CLUSTER to create RRSF work data sets. Although RACF allows a maximum FILESIZE value of 2147483647, the maximum value allowed for the RECORDS parameter by a particular release of Access Methods Services might be different. For example, DFSMS/MVS® Version 2 Release 4.0 restricts the maximum number of records to 16777215.

Note: The VSAM workspace data sets can be pre-allocated using the sample JCL provided in SAMPLIB member IRRSRRSF.

If you have pre-allocated the VSAM workspace data sets using the JCL provided in the IRRSRRSF SAMPLIB, those values remain in effect. Specifying values on the operands of the WORKSPACE keyword does not override those values. Values specified on the operands are displayed by the TARGET LIST NODE(*nodename*) command. Be aware that these values cannot be the actual values used for allocation.

Examples

Table 77. TARGET Examples

Example 1	<p><i>Operation</i> User ADMIN wants a summary list of the current operational attributes for all nodes in the RRSF configuration.</p> <p><i>Known</i> The RACF subsystem prefix is @.</p> <p><i>Command</i> @TARGET</p> <p><i>Defaults</i> Since no keywords are specified, LIST is the default. Since NODE is not specified, the command defaults to a summary listing for all target nodes known to the node the command runs on.</p> <p><i>Output</i> See Figure 63</p>
Example 2	<p><i>Operation</i> User ADMIN at POKMVSA wants to define POKMVSA as the local node for an RRSF configuration and make it operative.</p> <p><i>Known</i></p> <ul style="list-style-type: none"> • The RACF subsystem prefix is @. • POKMVSA has DFP non-SMS running. • The volume that contains the workspace data sets is DASD01. • The high-level qualifier for the workspace data sets is SYS1.RACF. • The APPC LUNAME is MF1AP001. • APPC and VTAM have been installed and configured. • Since the LIST keyword is specified in combination with other TARGET keywords, the information displayed provides the operational and protocol attributes as they exist after the processing of the other keywords. <p><i>Command</i> @TARGET NODE(POKMVSA) LOCAL DESCRIPTION('POUGHKEEPSIE MVS SYSTEM A') PREFIX(SYS1.RACF) WORKSPACE(VOLUME(DASD01)) PROTOCOL(APPC(LUNAME(MF1AP001))) OPERATIVE LIST</p> <p><i>Defaults</i></p> <ul style="list-style-type: none"> • The APPC TP name defaults to IRRRACF. • The modename defaults to the APPC default for the system. • FILESIZE defaults to 500. <p><i>Output</i> See Figure 63.</p>
Example 3	<p><i>Operation</i> User ADMIN wants to purge the OUTMSG workspace data set for node POKMVSB.</p> <p><i>Known</i> The RACF subsystem prefix is @. POKMVSB must be dormant to purge the workspace data sets. When the PURGE and DORMANT keywords are specified together, the DORMANT keyword is processed first.</p> <p><i>Command</i> @TARGET NODE(POKMVSB) PURGE(OUTMSG) DORMANT</p> <p><i>Defaults</i> None.</p> <p><i>Results</i> User ADMIN receives an informational message.</p> <pre>@IRRM021I RACF SUBSYSTEM PURGE OF NODE POKMVSB OUTMSG FILE SYS1.RACF.POKMVSA.OUTMSG IS COMPLETE. @IRRM002I RACF SUBSYSTEM TARGET COMMAND HAS COMPLETED SUCCESSFULLY.</pre>

TARGET

Table 77. TARGET Examples (continued)

Example 4 *Operation* User ADMIN wants to delete node POKMVSC from the set of known target nodes.
 Known The RACF subsystem prefix is @. POKMVSC is already dormant.
 Command @TARGET NODE(POKMVSC) DELETE
 Defaults None.

```
=====
Each form of this TARGET command gets the same output:
=====
- NODE1 <target
  or
- NODE1 <target list

00- NODE1 IRRM009I (<) LOCAL RRSF NODE NODE1 IS IN THE OPERATIVE ACTIVE
- STATE.
- IRRM009I (<) REMOTE RRSF NODE NODE2 IS IN THE OPERATIVE ACTIVE STATE
- IRRM009I (<) REMOTE RRSF NODE NODE3 IS IN THE OPERATIVE PENDING
- CONNECTION STATE.
- IRRM009I (<) REMOTE RRSF NODE NODE4 IS IN THE OPERATIVE PENDING
- CONNECTION STATE.
- IRRM009I (<) REMOTE RRSF NODE RSFNODE4 IS IN THE OPERATIVE PENDING
- CONNECTION STATE.
- IRRM009I (<) REMOTE RRSF NODE RSFNODE4 IS IN THE OPERATIVE PENDING
- CONNECTION STATE.

=====
target command for the local node
=====
<TARGET N(NODE1) LIST
IRRM010I (<) RSFJ SUBSYSTEM PROPERTIES OF LOCAL RRSF NODE NODE1:
STATE - OPERATIVE ACTIVE
DESCRIPTION - <NOT SPECIFIED>
PROTOCOL - APPC
          LU NAME - MF1AP001
          TP PROFILE NAME - IRRRACF
          MODENAME - <NOT SPECIFIED>
TIME OF LAST TRANSMISSION TO - <NONE>
TIME OF LAST TRANSMISSION FROM - <NONE>
WORKSPACE FILE SPECIFICATION
PREFIX - "RSFJ.WORK"
WDSQUAL - <NOT SPECIFIED>
FILESIZE - 500
VOLUME - TEMP01
FILE USAGE
          "RSFJ.WORK.NODE1.INMSG"
          - CONTAINS 0 RECORD(S)
          - OCCUPIES 1 EXTENT(S)
          "RSFJ.WORK.NODE1.OUTMSG"
          - CONTAINS 0 RECORD(S)
          - OCCUPIES 1 EXTENT(S)
```

Figure 63. Example 1: Output for TARGET Command (Part 1 of 2)

```

=====
target command for a remote node
=====
<TARGET N(NODE2) LIST
or
<TARGET N(NODE2)
IRRM010I (<) RSFJ SUBSYSTEM PROPERTIES OF REMOTE RRSF NODE NODE2:
  STATE - OPERATIVE ACTIVE
  DESCRIPTION - <NOT SPECIFIED>
  PROTOCOL - APPC
             LU NAME - MF2AP001
             TP PROFILE NAME - IRRRACF
             MODENAME - <NOT SPECIFIED>
  TIME OF LAST TRANSMISSION TO - <NONE>
  TIME OF LAST TRANSMISSION FROM - <NONE>
  WORKSPACE FILE SPECIFICATION
    PREFIX - "RSFJ.WORK"
    WDSQUAL - <NOT SPECIFIED>
    FILESIZE - 500
    VOLUME - TEMP01
    FILE USAGE
      "RSFJ.WORK.MF1AP001.MF2AP001.INMSG"
        - CONTAINS 0 RECORD(S)
        - OCCUPIES 1 EXTENT(S)
      "RSFJ.WORK.MF1AP001.MF2AP001.OUTMSG"
        - CONTAINS 0 RECORD(S)
        - OCCUPIES 1 EXTENT(S)

```

Figure 63. Example 1: Output for TARGET Command (Part 2 of 2)

Appendix A. Naming Considerations for Resource Profiles

Profile Definitions

In RACF, resource profiles contain a description of a resource, including the authorized users and the access authority of each user. Resource profiles can be discrete, generic, or, additionally for the DATASET class, fully-qualified generic. Regardless of whether a resource profile name is discrete or generic, the format of the name must follow certain rules. For profiles that protect general resources these rules are described by the entries in the class descriptor table and vary from class to class. For profiles that protect data sets these rules are the same as those used by TSO except that the high-level qualifier must be a valid RACF-defined user id or group name. Also, because the first qualifier is an id, RACF expects a data set name to have a minimum of two qualifiers. For a description of the TSO/E data set naming conventions, see *z/OS TSO/E User's Guide*.

Discrete Profiles

A *discrete* profile can protect a single resource that has unique security requirements. A discrete profile matches the name of the resource it protects and cannot exist independently of the resource. In the DATASET class, if you delete the resource, you delete the profile.

For example, a profile protecting a resource named SMITH.REXX.EXEC in class DATASET would protect the data set named SMITH.REXX.EXEC.

Generic Profiles

A *generic* profile can protect several resources that have a similar naming structure and security requirements. Specify generic characters in the profile name if you want to protect more than one resource with the same security requirements.

One or more of the following generic characters are allowed:

- Percent sign (%)
- Single asterisk (*)
- Double asterisk (**)
- Ampersand (&);

Notes:

1. The double asterisk (**) is not allowed with the DATASET class if enhanced generic naming (EGN) is inactive.
2. The ampersand (&) is only for general resource profile names and only if the RACFVARS class is active. Resource profiles can be created to protect resource names with *unlike* names. See *z/OS Security Server RACF Security Administrator's Guide* for more information.

For example, a profile protecting a resource named SMITH.* in class DATASET would protect all of SMITH's data sets that did not have a more specific profile defined (NOEGN is in effect).

Fully-Qualified Generic Profiles (DATASET class only)

A *fully-qualified* generic profile matches exactly the name of the data set it protects.

Naming Considerations for Resource Profiles

One reason to choose a fully-qualified generic profile for data set protection is that the profile is not deleted if the data set is deleted. If the data set is deleted and then re-created, the protection is there without creating another profile. Another reason is to protect multiple copies with one profile.

Determining RACF Protection

Although multiple generic profiles can match a general resource name, only the most specific profile actually protects it. For example, AB.CD*, AB.CD.**, and AB.**.CD all match the data set AB.CD, but AB.CD** protects the data set.

The best way to determine which profile is protecting a given resource is to use one of the list commands.

To find out what profile is protecting a general resource, enter the RLIST command:

```
RLIST class-name resource-name
```

which looks for a discrete profile. If none is found, and generic profile checking is in effect for the class, the generic profile which protects the resource is displayed.

To find out what profile is protecting your data set, enter:

```
LISTDSD DA('data-set-name')
```

which looks for a discrete profile. If none is found, and generic profile checking is in effect for the DATASET class, enter:

```
LISTDSD DA('data-set-name') GENERIC
```

which looks for a generic profile.

The rest of the appendix discusses the rules governing:

- Profile names for data sets
- Profile names for general resources.

Profile Names for Data Sets

Table 78 on page 496 shows the use of generic characters before RACF Release 1.9 and with RACF Release 1.9 or later. The use of the generic character % has not changed.

Note: Depending on whether EGN is active, the ending * has different meanings. These are explained in more detail later in this section.

Table 78. Generic Naming for Data Sets

Release of RACF	Ending **. Allowed	Middle **. Allowed	Beginning *. Allowed	Middle *. Allowed	Ending *. Allowed
Starting With RACF 1.9 EGN on	Yes	Yes	No	Yes	Yes
Starting With RACF 1.9 EGN off	No	No	No	Yes	Yes
Prior to RACF 1.9 EGN on	Yes	No	No	Yes	Yes
Prior to RACF 1.9 EGN off	No	No	No	Yes	Yes

Naming Considerations for Resource Profiles

For naming profiles in the DATASET class you can use discrete, generic, or fully-qualified generic names.

Discrete Profiles

These are the same as TSO data set names (see *z/OS TSO/E Command Reference*), except that the high-level qualifier (or the qualifier supplied by a command installation exit) must be a valid RACF-defined user ID or group name.

Generic Profile Rules—Enhanced Generic Naming Inactive

In the DATASET class, you can use generic characters as follows:

- Specify % to match any single character in a data set name
- Specify * as follows:
 - As a character at the end of a data set profile name (for example, ABC.DEF*) to match zero or more characters until the end of the name, zero or more qualifiers until the end of the data set name, or both
 - As a qualifier at the end of a profile name (for example, ABC.DEF.*) to match one or more qualifiers until the end of the data set name
 - As a qualifier in the middle of a profile name (for example, ABC.*.DEF) to match any one qualifier in a data set name
 - As a character at the end of a qualifier in the middle of a profile name (for example, ABC.DE*.FGH) to match zero or more characters until the end of the qualifier in a data set name.

Note: For profiles in the DATASET class, the high-level qualifier of the profile name must not be, nor can it contain, a generic character—for example, *.ABC, AB%.B, and AB*.AB are not allowed.

Tables are provided to show the variety of profiles that can be created using generics, using enhanced generic naming, and what happens to the profile protection if enhanced generic naming is turned off.

Table 79 and Table 80 provide examples of data set names using generic naming. Enhanced generic naming has not been turned on (SETRPTS NOEGN, the default, is in effect).

Table 81 and Table 82 provide examples of data set names with enhanced generic naming (SETR EGN is on).

Table 83 and Table 84 provide examples of data set names if enhanced generic naming is turned off after being turned on. It is not recommended that you turn EGN off after you have turned it on.

Table 79. Generic Naming for Data Sets with Enhanced Generic Naming Inactive—Asterisk at the End

Profile Name	AB.CD*	AB.CD.*
Resources protected by the profile	AB.CD AB.CDEF AB.CD.EF AB.CD.XY AB.CD.EF.GH	AB.CD.EF AB.CD.XY AB.CD.EF.GH

Naming Considerations for Resource Profiles

Table 79. Generic Naming for Data Sets with Enhanced Generic Naming Inactive—Asterisk at the End (continued)

Profile Name	AB.CD*	AB.CD.*
Resources not protected by the profile	ABC.DEF ABC.XY.XY.DEF	AB.CD AB.CDEF ABC.DEF AB.XY.XY.DEF

Table 80. Generic Naming for Data Sets with Enhanced Generic Naming Inactive—Asterisk in the Middle or %

Profile Name	ABC.%EF	AB.*.CD	AB.CD*.EF
Resources protected by the profile	ABC.DEF ABC.XEF	AB.CD.CD	AB.CDEF.EF AB.CDE.EF
Resources not protected by the profile	ABC.DEFGHI ABC.DEF.GHI ABC.DDEF	AB.CD AB.CD.EF AB.CDEF ABC.DEF ABC.XY.CD AB.XY.XY.CD	AB.CD.XY.EF

Generic Profile Rules—Enhanced Generic Naming Active

The *enhanced generic naming* option applies only to data sets and allows you to use double asterisk (**) in the DATASET class. It also changes the meaning of the single asterisk (*) at the end of a profile name.

Your RACF security administrator activates enhanced generic naming by issuing the SETROPTS command with the EGN operand. SETROPTS EGN makes the rules for data set and general resource profiles consistent with each other. Additionally, generic profiles can be more precise, and the generic profile names are more similar to other IBM products.

New installations should set EGN on immediately.

The following rules apply if you have enhanced generic naming in effect.

Specify * as follows:

- As a character at the end of a data set profile name to match zero or more characters until the end of the qualifier.
- As a qualifier at the end of a profile name to match *one* qualifier until the end of the data set name.

The meaning of an ending asterisk depends on whether the installation is using generic profiles with or without EGN.

Specify ** as follows:

- As either a middle or end qualifier in a profile name to match zero or more qualifiers. Only one occurrence of a double asterisk is allowed in a profile name. For example, ABC.DE.** is allowed; ABC.DE** is not allowed; and A.**.B.** is not allowed.

RACF does not allow you to specify any generic characters in the high-level qualifier of a data set name.

Naming Considerations for Resource Profiles

Table 81 and Table 82 show examples of generic profile names you can create when enhanced generic naming is active, and the resources protected and not protected by those profiles.

Table 81. Generic Data Set Profile Names Created with Enhanced Generic Naming Active—Asterisk and Double Asterisk at the End

Profile Name	AB.CD*	AB.CD.*	AB.CD.**	AB.CD*.**	AB.CD.*.**
Resources protected by the profile	AB.CD AB.CDEF	AB.CD.EF AB.CD.XY	AB.CD AB.CD.EF AB.CD.EF.GH AB.CD.XY	AB.CD AB.CD.EF AB.CDEF AB.CDEF.GH AB.CD.EF.GH AB.CD.XY	AB.CD.EF AB.CD.EF.GH AB.CD.XY
Resources not protected by the profile	AB.CD.EF AB.CD.EF.GH AB.CD.XY ABC.DEF	AB.CD AB.CDEF AB.CD.EF.GH ABC.DEF	AB.CDEF AB.CDE.FG ABC.DEF	ABC.DEF	ABC.DEF AB.CDEF AB.CDEF.GH AB.CD ABC.XY.XY.EF

Table 82. Generic Data Set Profile Names Created with Enhanced Generic Naming Active—Asterisk, Double Asterisk, or Percent Sign in the Middle

Profile Name	ABC.%EF	AB.*.CD	AB.**.CD
Resources protected by the profile	ABC.DEF ABC.XEF	AB.CD.CD	AB.CD AB.X.CD AB.X.Y.CD
Resources not protected by the profile	ABC.DEFGHI ABC.DEF.GHI ABC.DDEF	AB.CD AB.CD.EF AB.CDEF ABC.DEF ABC.XY.CD ABC.XY.XY.CD	AB.CD.EF AB.CDEF ABC.X.CD.EF ABC.DEF ABX.YCD

Note: Although multiple generic profiles might match a data set name, only the most specific actually protects the data set. For example, AB.CD*, AB.CD**, and AB.**.CD all match the data set AB.CD, but AB.CD** protects the data set.

In general, given two profiles that match a data set, you can find the more specific one by comparing the profile name from left to right. Where they differ, a non-generic character is more specific than a generic character. In comparing generics, a % is more specific than an *, and an * is more specific than **. Another way to determine the most specific is with the SEARCH command, as there are some rare exceptions to the general rule. SEARCH always lists the profiles in the order of the most specific to the least specific.

Data set profiles created before enhanced generic naming is activated continue to provide the same RACF protection after this option is activated.

If you protect resources with generic profiles while enhanced generic naming is active and then deactivate this option, your resources can no longer be protected. Table 83 on page 500 and Table 84 on page 500 show examples of generic profiles created with enhanced generic naming active and the protection after deactivation.

Naming Considerations for Resource Profiles

Table 83. After Deactivating EGN—Asterisk and Percent Sign in the Middle

Profile Name	ABC.%EF	ABC.*.DEF
How RACF displays the name after EGN is deactivated	ABC.%EF	ABC.*.DEF
Resources protected by the profile after EGN is deactivated	Same as before	Same as before

Table 84. After Deactivating EGN—Asterisk and Double Asterisk at the End

Profile Name	AB.CD*	AB.CD.*	AB.CD.**	AB.CD*.**	AB.CD.**.**
How RACF displays the name after EGN is deactivated	AB.CD*	AB.CD.*	AB.CD.	AB.CD*	AB.CD.*
Resources protected by the profile after EGN is deactivated	None	None	None	Same as before	Same as before

Choosing between Discrete and Generic Profiles

- Choose a *generic profile* for one of the following reasons:
 - To protect more than one data set with the same security requirements. The data sets protected by a generic profile must have some identical characters in their names. The profile name contains one or more generic characters (* or %).
 - If you have a single data set that might be deleted, then re-created, and you want the protection to remain the same, you can create a fully-qualified generic profile.
- Choose a *discrete profile* for the following reason:
 - To protect one data set with unique security requirements. The name of a discrete profile matches the name of the data set it protects.
While you could also use a fully-qualified generic, you should do so with care. Generic profiles can cause performance problems if they are not used to protect several data sets.

If a data set is protected by both a generic profile and a discrete profile, the discrete profile takes precedence and sets the level of protection for the data set.

If a data set is protected by more than one generic profile, the *most specific* profile takes precedence and sets the level of protection for the data set.

Notes:

1. All the members of a partitioned data set (PDS) are protected by one profile (the profile that protects the data set).
2. For a generic profile, unit and volume information is ignored because the data sets that are protected under the generic profile can be on many different volumes.

Profile Support

A generic profile might already exist under which the data set is protected. However, that profile might not provide the exact protection you want for your data set. In this case, you can create a more specific generic profile or a discrete profile for the data set.

All the components of a VSAM data set are protected by one profile (the profile that protects the cluster name). You do not need to create profiles that protect the index and data components of a cluster.

The protection offered by a generic profile is different, depending on the level of data management support installed on your system. Generic profiles protect all data sets that they apply to, including existing data sets and data sets to be created in the future.

A generic profile controls the *creation* of data sets. When a user creates a new tape or DASD data set that is protected by an existing generic profile, that profile must give the user ALTER authority. If the new data set is a group data set, the user must have either ALTER authority in the profile or CREATE authority in the group.

Data sets that are not RACF-indicated but are protected by a generic profile are *not protected* if they are transferred (in any way) or available (such as through shared DASD) to another system unless that system has all of the following:

- RACF protection
- The appropriate predefined generic profiles.

For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

Profile Names for General Resources

For the syntax required for profiles in the DIRECTRY and FILE classes, see the *RACF Command Language Reference* for your VM system. Table 85 shows the availability of generic characters before and with RACF Release 1.9 or later. The usage of the generic character % has not changed.

Note: The ending asterisk has different meanings and is explained further in the appropriate sections.

Table 85. Generic Naming for General Resources

	Double Asterisk Allowed in Beginning, Middle, or End	Middle Asterisk Allowed	Beginning Asterisk Allowed	Ending Asterisk Allowed
Starting With RACF 1.9	Yes	Yes	Yes	Yes
Prior to RACF 1.9	No	No	No	Yes

For naming general resources, you can use discrete or generic profiles. As mentioned before, discrete profile names exactly match the general resource name.

Valid generic characters are a percent sign (%), asterisk (*), double asterisk (**), and ampersand (&):

- Specify a percent sign to match any single character in a resource profile name.
- Specify a double asterisk once in a profile name as follows:

Naming Considerations for Resource Profiles

- As the entire profile name to match all resource names in a class.
- As either a beginning, middle, or ending qualifier (for example, `**ABC`, `ABC.**.DEF`, or `ABC.**`) to match zero or more qualifiers in a resource name.

Note: `**` is always available for general resources. The SETROPTS EGN setting is exclusively for data sets.

- Specify an asterisk as follows:
 - As a qualifier at the beginning of a profile name to match any one qualifier in a resource name.
 - As a character at the end of a profile name (for example, `ABC.DEF*`) to match zero or more characters until the end of the resource name, zero or more qualifiers until the end of the resource name, or both.
 - As a qualifier at the end of a profile name (for example, `ABC.DEF.*`) to match one or more qualifiers until the end of the resource name.
 - As a qualifier in the middle of a profile name (for example, `ABC.*.DEF`) to match any one qualifier in a resource name.
 - As a character at the end of a qualifier in the middle of a profile name (for example, `ABC.DE*.FGH`) to match zero or more characters until the end of the qualifier in a resource name.
- Specify an ampersand as follows:
 - In a profile name to indicate that RACF is to use a profile in the RACFVARS class to determine the actual values to use for that part of the profile name.

Note: When `KEYQUAL=nnn` is coded in the ICHERCDE macro, generic profiles created in that class can not contain generic characters in the first 'nnn' qualifiers of the profile.

See *z/OS Security Server RACF Security Administrator's Guide* for the unique naming conventions of specific classes and for a discussion of the RACFVARS class. See also the product documentation (such as PSF or CICS) for the naming conventions of specific classes.

Restricted Use of %* in General Resources

New profiles with an ending `%*` are not allowed, nor are profiles named `%*`. The RDEFINE command returns an error message.

Existing profiles with an ending `%*` are usable, but they should be deleted before creating any new profiles with a middle or beginning `*` or `**`. The RALTER and RDELETE commands accept `%*` to enable you to make the changes.

Instead of using an ending `%*`, create new profiles ending with `%.**` or `*.` for similar function (change `AB.C%*` to `AB.C%.**` or `AB.C*.`).

If you have existing profiles named `%*`, you should create new profiles (suggested name `**`).

Note: When creating the new profiles, consider using the FROM operand for continued use of the same access list.

Table 86, Table 87, and Table 88 give examples of generic profile names for general resources.

Naming Considerations for Resource Profiles

Table 86. Generic Naming for General Resources—Percent Sign, Asterisk, or Double Asterisk at the Beginning

Profile Name	% .AB	* .AB	** .AB
Resources protected by the profile	B .AB A .AB	AB .AB ABC .AB A .AB	AB A .A .A .AB AB .AB A .AB
Resources not protected by the profile	AB .AB ABC .AB	AB .CD AB .C .AB AB	ABC .AB .DEF ABAB

Table 87. Generic Naming for General Resources—Asterisk or Double Asterisk at the Ending

Profile Name	AB .CD*	AB .CD .*	AB .CD .**
Resources protected by the profile	AB .CD AB .CDEF AB .CD .EF AB .CD .XY AB .CD .EF .GH	AB .CD .EF AB .CD .XY AB .CD .EF .XY	AB .CD .CD AB .CD .X .Y .Z AB .CD AB .CD .EF .GH
Resources not protected by the profile	ABC .DEF ABC .XY .XY .DEF	AB .CD AB .CDEF ABC .DEF AB .XY .XY .DEF	ABC .CD AB .CDE .EF

Table 88. Generic Naming for General Resources—Asterisk, Double Asterisk, or Percent Sign in the Middle

Profile Name	ABC .%EF	AB .* .CD	AB .CD* .CD	AB .** .CD
Resources protected by the profile	ABC .DEF ABC .XEF	AB .CD .CD	AB .CD .CD AB .CDEF .CD	AB .CD AB .X .CD AB .X .Y .CD
Resources not protected by the profile	ABC .DEFGHI ABC .DEF .GHI	AB .CD AB .CD .EF AB .CDEF AB .X .Y .CD	AB .CD .XY AB .CD .XY .CD	AB .CD .EF AB .CDEF ABC .X .CD .EF ABC .DEF ABC .XY .CD ABC .XY .XY .CD

Although multiple generic profiles might match a general resource name, only the most specific actually protects the resource. For example, AB.CD*, AB.CD**, and AB.**.CD all match the general resource AB.CD, but AB.CD** protects it.

In general, given two profiles that match a general resource, you can find the more specific one by comparing the profile name from left to right. Where they differ, a nongeneric character is more specific than a generic character. In comparing generics, a percent sign is more specific than an asterisk, and an asterisk is more specific than double asterisk. Another way to determine the most specific is with the SEARCH command, as there are some rare exceptions to the general rule. SEARCH always lists the profiles in the order of the most specific to the least specific.

Permitting Profiles for GENERICOWNER Classes

GENERICOWNER gives an installation the ability of restricting CLAUTH users from creating profiles in a class. In order to do this, a top-level ** profile is defined. This profile is owned by the system administrator and this profile blocks all non-SPECIAL

Naming Considerations for Resource Profiles

users from creating profiles. A *permitting profile* must be defined for each CLAUTH user. Each profile defines the subset of resources in the class that the user is allowed to create.

When a CLAUTH user attempts to define a resource, a search is made for a less-specific (permitting) profile that covers the profile being defined. This less-specific profile is a profile that matches the more-specific profile name, character for character, up to the ending * or ** or ending contiguous %(s) in the less-specific name.

This definition might appear simple, but is not exactly what you might expect in comparison to the preceding section.

Table 89. Permitting Profiles

Profile Name	AA.*	AA.**	AA*	A.*.B.**
covered	AA.BB AA.B.C AA.%%	AA.* AA AA.BB AA.B.C AA.%	AA.* AA AA.BB AA.B.C AAC.BB AA%.%%	A.*.B.CC A.*.B.%.%%
not covered	AA.** AA ABC.BB A%.AA	AAC.BB ABC.BB %A.%	ABC.BB A%A	A.A.B.CC A.%.B.%.%%

Table 90. Permitting Profiles (continued)

Profile Name	AA.%	AA.%%	AA%	A.*.B.%%
covered	AA.B	AA.BB AA.%B	AAC	A.*.B.CC
not covered	AA.** A%.A AA.CC	AA.B	AA.B A%A	A.A.B.CC A.%.B.%%

Commands to Administer VM Shared File System Profiles

In z/OS, you cannot use the 12 RACF commands that were added to RACF 1.10 for VM to manipulate shared file system (SFS) FILE and DIRECTORY profiles. You can, however, use the existing general resource commands (RDEFINE, RALTER, RDELETE, RLIST, PERMIT, and SEARCH), to manipulate FILE and DIRECTORY profiles.

The profile name that you enter on the general resource commands must be in RACF format. The RACF format name contains periods as separators between all qualifiers (no colon), and no punctuation following the last qualifier. For FILES, the file name and file type are the last two qualifiers in the name.

The following table summarizes the commands for administering SFS FILE and DIRECTORY profiles.

Table 91. Commands to Administer SFS Profiles

Function	Command to be used on OS/390 R3 and z/OS R1	Command provided on RACF 1.10 for VM
Add a DIRECTORY profile	RDEFINE DIRECTORY	ADDDIR ADIR

Naming Considerations for Resource Profiles

Table 91. Commands to Administer SFS Profiles (continued)

Function	Command to be used on OS/390 R3 and z/OS R1	Command provided on RACF 1.10 for VM
Add a FILE profile	RDEFINE FILE	ADDFILE AF
Alter a DIRECTORY profile	RALTER DIRECTORY	ALTDIR
Alter a FILE profile	RALTER FILE	ALTFILE ALF
Delete a DIRECTORY profile	RDELETE DIRECTORY	DELDIR DDIR
Delete a FILE profile	RDELETE FILE	DELFILE DF
List a DIRECTORY profile	RLIST DIRECTORY	LDIRECT LDIR
List a FILE profile	RLIST FILE	LFILE LF
Change access list in a DIRECTORY profile	PERMIT <i>profile-name</i> CLASS(DIRECTORY)	PERMDIR PDIR
Change access list in a FILE profile	PERMIT <i>profile-name</i> CLASS(FILE)	PERMFILE PF
Search for a DIRECTORY profile	SEARCH CLASS(DIRECTORY)	SRDIR
Search for a FILE profile	SEARCH CLASS(FILE)	SRFILE SRF

Naming Considerations for Resource Profiles

Appendix B. Description of RACF Classes

Describing RACF classes

The following sections describe the general resource classes you can find in the class descriptor table (CDT). See *z/OS Security Server RACF Macros and Interfaces* to find details (such as POSIT values) for each class.

Supplied resource classes for z/OS and OS/390 systems

Table 92 lists the supplied classes that can be used on z/OS and OS/390 systems.

Table 92. Resource Classes for z/OS and OS/390 Systems

Class Name	Description
ALCSAUTH	Supports the Airline Control System/MVS (ALCS/MVS) product.
APPCLU	Verifying the identity of partner logical units during VTAM session establishment.
APPCPORT	Controlling which user IDs can access the system from a given LU (APPC port of entry). Also, conditional access to resources for users entering the system from a given LU.
APPCSERV	Controlling whether a program being run by a user can act as a server for a specific APPC transaction program (TP).
APPCSI	Controlling access to APPC side information files.
APPCTP	Controlling the use of APPC transaction programs.
APPL	Controlling access to applications.
CACHECLS	Contains profiles used for saving and restoring cache contents from the RACF database.
CBIND	Controlling the client's ability to bind to the server.
CONSOLE	Controlling access to MCS consoles. Also, conditional access to other resources for commands originating from an MCS console.
CSFKEYS	Controlling use of Integrated Cryptographics Service Facility (ICSF) cryptographic keys. See also the GCSFKEYS class.
CSFSERV	Controlling use of Integrated Cryptographics Service Facility (ICSF) cryptographic services.
DASDVOL	DASD volumes. See also the GDASDVOL class.
DBNFORM	Reserved for future IBM use.
DEVICES	Used by MVS allocation to control who can allocate devices such as: <ul style="list-style-type: none">• Unit record devices (printers and punches) (allocated only by PSF, JES2, or JES3)• Graphics devices (allocated only by VTAM)• Teleprocessing (TP) or communications devices (allocated only by VTAM)
DIGTCERT	Contains digital certificates and information related to them.
DIGTCRIT	Specifies additional criteria for certificate name filters.
DIGTNMAP	Mapping class for certificate name filters.
DIGTRING	Contains a profile for each key ring and provides information about the digital certificates that are part of each key ring.

RACF Classes

Table 92. Resource Classes for z/OS and OS/390 Systems (continued)

Class Name	Description
DIRAUTH	Setting logging options for RACROUTE REQUEST=DIRAUTH requests. Also, if the DIRAUTH class is active, security label authorization checking is done when a user receives a message sent through the TPUT macro or the TSO SEND, or LISTBC commands. Profiles are not allowed in this class.
DLFCLASS	The data lookaside facility.
FACILITY	Miscellaneous uses. Profiles are defined in this class so that resource managers (typically program products or components of MVS or VM) can check a user's access to the profiles when the users take some action. Examples are catalog operations (DFP) and use of the vector facility (an MVS component). RACF does not document all of the resources used in the FACILITY class by other products. For information on the FACILITY class resources used by a specific product (other than RACF itself), see the product's documentation.
FIELD	Fields in RACF profiles (field-level access checking).
GCSFKEYS	Resource group class for CSFKEYS class. ¹
GDASDVOL	Resource group class for DASDVOL class. ¹
GLOBAL	Global access checking table entry. ¹
GMBR	Member class for GLOBAL class (not for use on RACF commands).
GSDSF	Resource group class for SDSF class. ¹
GTERMINL	Resource group class for TERMINAL class. ¹
IBMOPC	Controlling access to OPC/ESA subsystems.
JESINPUT	Conditional access support for commands or jobs entered into the system through a JES input device.
JESJOBS	Controlling the submission and cancellation of jobs by job name.
JESSPOOL	Controlling access to job data sets on the JES spool (that is, SYSIN and SYSOUT data sets).
KEYSMSTR	Contains profiles that hold keys to encrypt data stored in the RACF database, such as LDAP BIND passwords and DCE passwords.
LDAPBIND	Contains the LDAP server URL, bind distinguished name, and bind password.
LOGSTRM	Reserved for MVS/ESA.
NODES	Controlling the following on MVS systems: <ul style="list-style-type: none"> • Whether jobs are allowed to enter the system from other nodes • Whether jobs that enter the system from other nodes have to pass user identification and password verification checks
NODMBR	Member class for NODES class (not for use on RACF commands).
OPERCMDS	Controlling who can issue operator commands (for example, JES and MVS, and operator commands). ²
PMBR	Member class for PROGRAM class (not for use on RACF commands).
PROGRAM	Controlled programs (load modules). ¹
PROPCNTL	Controlling if user ID propagation can occur, and if so, for which user IDs (such as the CICS or IMS main task user ID), user ID propagation is <i>not</i> to occur.

Table 92. Resource Classes for z/OS and OS/390 Systems (continued)

Class Name	Description
PSFMPL	Used by PSF to perform security functions for printing, such as separator page labeling, data page labeling, and enforcement of the user printable area.
PTKTDATA	PassTicket key class enables the security administrator to associate a RACF secured signon secret key with a particular mainframe application that uses RACF for user authentication. Examples of such applications are IMS, CICS, TSO, VM, APPC, and MVS batch.
RACGLIST	Class of profiles that hold the results of RACROUTE REQUEST=LIST,GLOBAL=YES or a SETROPTS RACLIST operation.
RACFVARS	RACF variables. In this class, profile names, which start with & (ampersand), act as RACF variables that can be specified in profile names in other RACF general resource classes.
RRSFDATA	Used to control RACF remote sharing facility functions.
RVARSMBR	Member class for RACFVARS (not for use on RACF commands).
SCDMBR	Member class for SECDATA class (not for use on RACF commands).
SDSF	Controls the use of authorized commands in the System Display and Search Facility (SDSF). See also GSDSF class.
SECDATA	Security classification of users and data (security levels and security categories). ¹
SECLABEL	If security labels are used, and, if so, their definitions. ²
SERVAUTH	Contains profiles that are used by servers to check a client's authorization to use the server or to use resources managed by the server.
SERVER	Controlling the server's ability to register with the daemon.
SMESSAGE	Controlling to which users a user can send messages (TSO only).
SOMDOBS	Controlling the client's ability to invoke the method in the class.
STARTED	Used in preference to the started procedures table to assign an identity during the processing of an MVS START command.
SURROGAT	If surrogate submission is allowed, and if allowed, which user IDs can act as surrogates.
SYSMVIEW	Controlling access by the SystemView for MVS Launch Window to SystemView for MVS applications.
TAPEVOL	Tape volumes.
TEMPDSN	Controlling who can access residual temporary data sets. You cannot create profiles in this resource class.
TERMINAL	Terminals (TSO or VM). See also GTERMINL class.
VTAMAPPL	Controlling who can open ACBs from non-APF authorized programs.
WRITER	Controlling the use of JES writers.
CICS classes	
ACICSPCT	CICS program control table. ²
BCICSPCT	Resource group class for ACICSPCT class. ¹
CCICSCMD	Used by CICS/ESA 3.1, or later, to verify that a user is permitted to use CICS system programmer commands such as INQUIRE, SET, PERFORM, and COLLECT. ¹

RACF Classes

Table 92. Resource Classes for z/OS and OS/390 Systems (continued)

Class Name	Description
CPSMOBJ	Used by CICSplex System Manager, which provides a central point of control when running multiple CICS systems, to determine operational controls within a CICS complex.
CPSMXMP	Used by CICSplex System Manager to identify exemptions from security controls within a CICS complex.
DCICSDCT	CICS destination control table. ²
ECICSDCT	Resource group class for DCICSDCT class. ¹
FCICSFCT	CICS file control table. ²
GCICSTRN	Resource group class for TCICSTRN class. ²
GCPSMOBJ	Resource grouping class for CPSMOBJ.
HCICSFCT	Resource group class for FCICSFCT class. ¹
JCICSJCT	CICS journal control table. ²
KCICSJCT	Resource group class for JCICSJCT class. ¹
MCICSPPT	CICS processing program table. ²
NCICSPPT	Resource group class for MCICSPPT class. ¹
PCICSPSB	CICS program specification blocks or PSBs
QCICSPSB	Resource group class for PCICSPSB class. ¹
SCICSTST	CICS temporary storage table. ²
TCICSTRN	CICS transactions.
UCICSTST	Resource group class for SCICSTST class. ¹
VCICSCMD	Resource group class for the CCICSCMD class. ¹
DB2 classes	
DSNADM	DB2 administrative authority class.
DSNR	Controls access to DB2 subsystems.
GDSNBP	Grouping class for DB2 buffer pool privileges.
GDSNCL	Grouping class for DB2 collection privileges.
GDSNDB	Grouping class for DB2 database privileges.
GDSNJR	Grouping class for Java archive files (JARs).
GDSNPK	Grouping class for DB2 package privileges.
GDSNPN	Grouping class for DB2 plan privileges.
GDSNSC	Grouping class for DB2 schemas privileges.
GDSNSG	Grouping class for DB2 storage group privileges.
GDSNSM	Grouping class for DB2 system privileges.
GDSNSP	Grouping class for DB2 stored procedure privileges.
GDSNTB	Grouping class for DB2 table, index, or view privileges.
GDSNTS	Grouping class for DB2 tablespace privileges.
GDSNUF	Grouping class for DB2 user-defined function privileges.
GDSNUT	Grouping class for DB2 user-defined distinct type privileges.
MDSNBP	Member class for DB2 buffer pool privileges.
MDSNCL	Member class for DB2 collection privileges.
MDSNDB	Member class for DB2 database privileges.

Table 92. Resource Classes for z/OS and OS/390 Systems (continued)

Class Name	Description
MDSNJR	Member class for Java archive files (JARs).
MDSNPK	Member class for DB2 package privileges.
MDSNPN	Member class for DB2 plan privileges.
MDSNSC	Member class for DB2 schema privileges.
MDSNSG	Member class for DB2 storage group privileges.
MDSNSM	Member class for DB2 system privileges.
MDSNSP	Member class for DB2 stored procedure privileges.
MDSNTB	Member class for DB2 table, index, or view privileges.
MDSNTS	Member class for DB2 tablespace privileges.
MDSNUF	Member class for DB2 user-defined function privileges.
MDSNUT	Member class for DB2 user-defined distinct type privileges.
DCE classes	
DCEUUIDS	Used to define the mapping between a user's RACF user ID and the corresponding DCE principal UUID.
Enterprise Java Beans classes	
EJBROLE	Member class for Enterprise Java Beans authorization roles.
GEJBROLE	Grouping class for Enterprise Java Beans authorization roles.
JAVA	Contains profiles that are used by Java for z/OS applications to perform authorization checking for Java for z/OS resources.
IMS classes	
AIMS	Application group names (AGN).
CIMS	Command.
DIMS	Grouping class for command.
FIMS	Field (in data segment).
GIMS	Grouping class for transaction.
HIMS	Grouping class for field.
OIMS	Other.
PIMS	Database.
QIMS	Grouping class for database.
SIMS	Segment (in database).
TIMS	Transaction (trancode).
UIMS	Grouping class for segment.
WIMS	Grouping class for other.
Information/Management (Tivoli Service Desk) classes	
GINFOMAN	Grouping class for Information/Management (Tivoli Service Desk) resources.
INFOMAN	Member class for Information/Management (Tivoli Service Desk) resources.
Infoprint Server class	
PRINTSRV	Controls access to printer definitions for Infoprint Server.
LFS/ESA classes	

RACF Classes

Table 92. Resource Classes for z/OS and OS/390 Systems (continued)

Class Name	Description
LFSCCLASS	Controls access to file services provided by LFS/ESA.
License Manager class	
ILMADMIN	Controls access to the administrative functions of IBM License Manager.
Lotus Notes for z/OS and Novell Directory Services for OS/390 classes	
NDSLINK	Mapping class for Novell Directory Services for OS/390 user identities.
NOTELINK	Mapping class for Lotus Notes for z/OS user identities.
MQSeries classes	
GMQADMIN	Grouping class for MQSeries administrative options. ¹
GMQCHAN	Reserved for MQSeries.
GMQNLIST	Grouping class for MQSeries namelists. ¹
GMQPROC	Grouping class for MQSeries processes. ¹
GMQQUEUE	Grouping class for MQSeries queues. ¹
MQADMIN	Protects MQSeries administrative options.
MQCHAN	Reserved for MQSeries.
MQCMDS	Protects MQSeries commands.
MQCONN	Protects MQSeries connections.
MQNLIST	Protects MQSeries namelists.
MQPROC	Protects MQSeries processes.
MQQUEUE	Protects MQSeries queues.
NetView classes	
NETCMDS	Controlling which NetView commands the NetView operator can issue.
NETSPAN	Controlling which NetView commands the NetView operator can issue against the resources in this span.
NVASAPDT	NetView/Access Services.
PTKTVAL	Used by NetView/Access Services Secured Single Signon to store information needed when generating a PassTicket.
RMTOPS	NetView Remote Operations.
RODMMGR	NetView Resource Object Data Manager (RODM).
Network Authentication Service classes	
KERBLINK	Mapping class for user identities of local and foreign principals.
REALM	Used to define the local and foreign realms.
SMS (DFSMSdfp) classes	
MGMTCLAS	SMS management classes.
STORCLAS	SMS storage classes.
SUBSYSNM	Authorizes a subsystem (such as a particular instance of CICS) to open a VSAM ACB and use VSAM Record Level Sharing (RLS) functions.
Tivoli classes	
ROLE	Specifies the complete list of resources and associated access levels that are required to perform the particular job function this role represents and defines which RACF groups are associated with this role.
TMEADMIN	Maps the user IDs of Tivoli administrators to RACF user IDs.

Table 92. Resource Classes for z/OS and OS/390 Systems (continued)

Class Name	Description
TSO classes	
ACCTNUM	TSO account numbers.
PERFGRP	TSO performance groups.
TSOAUTH	TSO user authorities such as OPER and MOUNT.
TSOPROC	TSO logon procedures.
z/OS UNIX classes	
DIRACC	Controls auditing (using SETROPTS LOGOPTIONS) for access checks for read/write access to HFS directories. Profiles are not allowed in this class.
DIRSRCH	Controls auditing (using SETROPTS LOGOPTIONS) of HFS directory searches. Profiles are not allowed in this class.
FSOBJ	Controls auditing (using SETROPTS LOGOPTIONS) for all access checks for HFS objects except directory searches. Controls auditing (using SETROPTS AUDIT) of creation and deletion of HFS objects. Profiles are not allowed in this class.
FSSEC	Controls auditing (using SETROPTS LOGOPTIONS) for changes to the security data (FSP) for HFS objects. Profiles are not allowed in this class.
IPCOBJ	Controlling auditing and logging of IPC security checks.
PROCACT	Controls auditing (using SETROPTS LOGOPTIONS) of functions that look at data from, or affect the processing of, z/OS UNIX processes. Profiles are not allowed in this class.
PROCESS	Controls auditing (using SETROPTS LOGOPTIONS) of changes to UIDs and GIDs of z/OS UNIX processes. Controls auditing (using SETROPTS AUDIT) of dubbing and undubbing of z/OS UNIX processes. Profiles are not allowed in this class.
UNIXMAP	Contains profiles that are used to map z/OS UNIX UIDs to RACF user IDs and z/OS UNIX GIDs to RACF group names.
UNIXPRIV	Contains profiles that are used to grant z/OS UNIX privileges.
Notes:	
1. You cannot specify this class name on the GENCMD, GENERIC, and GLOBAL/NOGLOBAL operands of the SETROPTS command.	
2. You cannot specify this class name on the GLOBAL operand of SETROPTS or, if you do, the GLOBAL checking is not performed.	

Supplied resource classes for z/VM and VM systems

Table 93 lists the supplied classes you can use on z/VM and VM systems. These classes are primarily relevant if you share your RACF database with a z/VM or VM system.

Table 93. Resource Classes for z/VM and VM Systems

Class Name	Description
DIRECTRY	Protection of shared file system (SFS) directories.

RACF Classes

Table 93. Resource Classes for z/VM and VM Systems (continued)

Class Name	Description
FACILITY	Miscellaneous uses. Profiles are defined in this class so resource managers (typically program products or components of MVS or VM) can check a user's access to the profiles when the users take some action. Examples are using combinations of options for tape mounts, and use of the RACROUTE interface. RACF does not document all of the resources used in the FACILITY class by other products. For information on the FACILITY class resources used by a specific product (other than RACF itself), see that product's documentation.
FIELD	Fields in RACF profiles (field-level access checking).
FILE	Protection of shared file system (SFS) files.
GLOBAL	Global access checking. ¹
GMBR	Member class for GLOBAL class (not for use on RACF commands).
GTERMINL	Terminals whose IDs do not fit into generic profile naming conventions. ¹
PSFMPL	When class is active, PSF/VM performs separator and data page labeling as well as auditing.
PTKTDATA	PassTicket key class.
PTKTVAL	Used by NetView/Access Services Secured Single Signon to store information needed when generating a PassTicket.
RACFVARS	RACF variables. In this class, profile names, which start with & (ampersand), act as RACF variables that can be specified in profile names in other RACF general resource classes.
RVARSMBR	Member class for RACFVARS (not for use on RACF commands).
SCDMBR	Member class for SECDATA class (not for use on RACF commands).
SECDATA	Security classification of users and data (security levels and security categories). ¹
SECLABEL	If security labels are used and, if so, their definitions. ²
SFSCMD	Controls the use of shared file system (SFS) administrator and operator commands.
TAPEVOL	Tape volumes.
TERMINAL	Terminals (TSO or VM). See also GTERMINL class.
VMBATCH	Alternate user IDs.
VMBR	Member class for VMEVENT class (not for use on RACF commands).
VMCMD	Certain CP commands and other requests on VM.
VMEVENT	Auditing and controlling security-related events (called VM events) on VM systems.
VMMAC	Used in conjunction with the SECLABEL class to provide security label authorization for some VM events. Profiles are not allowed in this class.
VMMDISK	VM minidisks.
VMNODE	RSCS nodes.
VMRDR	VM unit record devices (virtual reader, virtual printer, and virtual punch).
VMSEGMT	Restricted segments, which can be named saved segments (NSS) and discontinuous saved segments (DCSS).
VXMBR	Member class for VMXEVENT class (not for use on RACF commands).

Table 93. Resource Classes for z/VM and VM Systems (continued)

Class Name	Description
VMXEVENT	Auditing and controlling security-related events (called VM events) on VM systems.
VMPOSIX	Contains profiles used by OpenExtensions VM.
WRITER	VM print devices.

Notes:

1. You cannot specify this class name on the GENCMD, GENERIC, and GLOBAL/NOGLOBAL operands of the SETROPTS command.
2. You cannot specify this class name on the GLOBAL operand of the SETROPTS command or, if you do, the GLOBAL checking is not performed.

See *z/OS Security Server RACF Macros and Interfaces* or *z/OS Security Server RACROUTE Macro Reference* for more information on the class descriptor table supplied by IBM.

Appendix C. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen-readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen-readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Volume I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

Notices

This information was developed for products and services offered in the USA. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs

Notices

and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

This product contains code licensed from RSA Data Security Incorporated.



Trademarks

The following terms are trademarks of the IBM Corporation in the United States, other countries, or both:

- BookManager
- CICS
- CICS/ESA

- CICSplex
- DB2
- DFSMS/MVS
- DFSMSdss
- DFSMSdfp
- Hiperbatch
- IBM
- IBMLink
- IMS
- Language Environment
- Library Reader
- MQSeries
- MVS
- MVS/DFP
- MVS/ESA
- OS/390
- Print Services Facility
- RACF
- Redbooks
- Resource Link
- S/390
- SecureWay
- System/390
- SystemView
- TalkLink
- VM/ESA
- VTAM
- z/Architecture
- z/OS
- z/OS.e
- z/VM
- zSeries

Lotus and Lotus Notes are trademarks of Lotus Development Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

NetView, TME, and Tivoli are trademarks of International Business Machines Corporation or Tivoli Systems Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Index

Numerics

64-bit Data Encryption Standard
SESSKEY operand 325, 361

A

access attempt
 changing
 for data set profile 95, 97
access authority
 changing in resource profiles 251
 data sets 13
access checking
 field-level 343
 list-of-groups 452
access list
 conditional 247
 copying from another profile 252
 deleting from profile 253
 deleting names from 251
 displaying for data set profile 204
 displaying for general resource profile 388
 standard 247
ACCESS operand
 PERMIT command 251
access to system
 controlling
 for existing user 166
 for new user 82
 restoring for user 158, 177
 revoking for user 158, 178
 terminals 333, 366
accessibility 517
account number for TSO
 changing for user 164
 for existing user 160, 161, 164, 165
 for new user 79
ACCTNUM class
 description 513
ACCTNUM suboperand
 ADDUSER command 79
 ALTUSER command 160
ACICSPCT class
 description 509
activating
 general resource classes 444
 JES options 453
 RACF system-wide options 435
ACTIVE operand
 RVARY command 401
ADD operand
 RACDCERT command 263
ADDCATEGORY operand
 ADDSD command 37
 ADDUSER command 53
 ALTDSD command 93
 ALTUSER command 121
ADDCATEGORY operand (*continued*)
 RALTER command 308
 RDEFINE command 343
ADDCREATOR operand
 SETROPTS command 440
ADDDIR 1
ADDDOMAINS suboperand
 ALTUSER command 137
ADDFILE 1
ADDGROUP command
 description 24
 examples 31
 RACF requirements 24
 syntax 24
ADDMEM operand
 RALTER command 309
 RDEFINE command 344
ADDMSCOPE suboperand
 ALTUSER command 150
ADDOPCLASS suboperand
 ALTUSER command 123, 138
address lines
 ALTUSER command 83
ADDRING operand
 RACDCERT command 281
ADDSD command
 description 33
 examples 45
 RACF requirements 34
 syntax 35
ADDUSER command
 description 48
 examples 85
 RACF requirements 49
 syntax 50
ADDVOL operand
 ALTDSD command 94
 RALTER command 311
administration
 classroom courses xii
ADSP (automatic data set protection) attribute
 activating or deactivating system-wide 441
 adding to user profile 122
 deleting from user profile 122
 for new user 53
 in user's connect profile 175
ADSP operand
 ADDUSER command 53
 ALTUSER command 122
 CONNECT command 175
 SETROPTS command 441
AGE operand
 SEARCH command 411
AIMS class
 description 511
ALCSAUTH class
 description 507

- alias data set name
 - RACF restriction on using 16
- ALL operand
 - HELP command 199
 - LISTDSD command 204
 - RLIST command 388
 - SEARCH command 411
- ALTDIR 1
- ALTDSD command
 - description 90
 - examples 102
 - RACF requirements 91
 - syntax 92
- ALTER operand
 - RACDCERT command 269
- ALTFILE 1
- ALTGROUP command
 - description 104
 - examples 113
 - RACF requirements 104
 - syntax 105
- ALTGRP suboperand
 - ADDUSER command 70
 - ALTUSER command 146
- ALTMAP operand
 - RACDCERT command 286
- ALTNAME operand
 - RACDCERT command 276
- ALTUSER command
 - changing
 - in user profile 130
 - description 115
 - examples 169
 - operator information 146
 - primary language 134
 - RACF requirements 116
 - secondary language 134
 - syntax 117
- ALTVOL operand
 - ALTDSD command 94
- APPCLU class
 - description 507
- APPCPORT class
 - description 507
- APPCSERV class
 - description 507
- APPCSI class
 - description 507
- APPCTP class
 - description 507
- APPL class
 - description 507
- APPL operand
 - DISPLAY command 194
 - SIGNOFF command 477
- APPLAUDIT operand
 - SETROPTS command 441
- APPLDATA operand
 - RALTER command 312
 - RDEFINE command 350
- application data
 - bypassing replay protection 312
 - changing for general resource profile 312
 - defining for general resource profile 350
 - deleting
 - from general resource profile 313
- application key
 - encrypting 326, 361
 - masking 326, 361
- ASSIZEMAX operand
 - ADDUSER command 65
- ASSIZEMAX suboperand
 - ALTUSER command 139
- AT operand
 - ADDGROUP command 26
 - ADDSD command 38
 - ADDUSER command 53
 - ALTDSD command 95
 - ALTGROUP command 107
 - ALTUSER command 122
 - CONNECT command 175
 - DELDSD command 183
 - DELGROUP command 188
 - DELUSER command 191
 - LISTDSD command 204
 - LISTGRP command 217
 - LISTUSER command 227
 - PASSWORD command 244
 - PERMIT command 250
 - RALTER command 313
 - RDEFINE command 351
 - RDELETE command 374
 - REMOVE command 378
 - RLIST command 388
 - SEARCH command 411
 - SETROPTS command 442
- attribute
 - AUDITOR
 - for existing user 122
 - for new user 53
 - CLAUTH (class authority)
 - for existing user 124
 - for new user 55
 - EIM (Enterprise Identity Mapping)
 - for existing user 129
 - for new user 59
 - group-AUDITOR
 - in user's connect profile 176
 - group-OPERATIONS
 - in user's connect profile 177
 - logging activities for 460
 - group-SPECIAL
 - in user's connect profile 179
 - logging activities for 468
 - GRPACC (group access)
 - for existing user 130
 - for new user 59
 - in user's connect profile 176
 - OPERATIONS
 - for existing user 145
 - for new user 69

- attribute (*continued*)
 - OPERATIONS (*continued*)
 - logging activities for 460
 - SPECIAL
 - for existing user 160
 - for new user 79
 - logging activities for 468
- audit access level
 - adding to new data set profile 38
 - changing
 - for data set profile 95, 97
 - changing for general resource profile 314, 316
 - defining for general resource profile 351
- AUDIT operand
 - ADDSD command 38
 - ALTDSD command 95
 - RALTER command 313
 - RDEFINE command 351
 - SETROPTS command 442
- AUDITOR attribute
 - for existing user 122
 - for new user 53
- AUDITOR operand
 - ADDUSER command 53
 - ALTUSER command 122
 - CONNECT command 176
- AUTH suboperand
 - ADDUSER command 70
 - ALTUSER command 146
- authority
 - required to issue RACF commands 1
 - summary 1
- AUTHORITY operand
 - ADDUSER command 54
 - ALTUSER command 122
 - CONNECT command 176
- AUTHUSER operand
 - LISTDSD command 205
 - RLIST command 388
- AUTO request reception
 - delete from user profile 147
- AUTO suboperand
 - ADDUSER command 70
 - ALTUSER command 147
- AUTOAPPL operand
 - SET command 422
- AUTODIRECT operand
 - SET command 422
- AUTOGID suboperand
 - ALTGROUP command 109
- automatic data protection (ADSP) attribute
 - See ADSP (automatic data set protection) attribute
- automatic direction of application updates 422
- automatic TAPEVOL profile
 - altering 303
 - permitting access to 248
- AUTOUID suboperand
 - ADDUSER command 65
 - ALTUSER command 139

B

- batch
 - submitting RACF TSO commands from 16
- BCICSPCT class
 - description 509
- BINDDN suboperand
 - ADDUSER command 77
 - ALTUSER command 156
 - RALTER command 322
 - RDEFINE command 358
- BINDPW suboperand
 - ADDUSER command 77
 - ALTUSER command 157
 - RALTER command 322
 - RDEFINE command 358
- bypassing
 - recording of statistics 453

C

- CACHECLS class
 - description 507
- canceling
 - syntax rules for passwords 462
 - system-wide ADSP (automatic data set protection)
 - attribute 441
- CATDSNS operand
 - SETROPTS command 443
- CATEGORY operand
 - SEARCH command 412
- CBIND class
 - description 507
- CCICSCMD class
 - description 509
- CERTAITH operand
 - RACDCERT command 261
- CERTSIGN operand
 - RACDCERT command 275
- changing
 - for existing data set profile 329
 - password interval 244
- CHECKCERT operand
 - RACDCERT command 269
- choosing between discrete and generic profiles
 - using commands on MVS 500
- CICS
 - general resource classes 509
- CICS operand
 - ADDUSER command 54
 - ALTUSER command 123
 - LISTUSER command 227
- CICS segment
 - user profile
 - defining 54
 - displaying 227
- CIMS class
 - description 511
- class descriptor table (CDT)
 - protecting classes 1
 - supplied classes for z/OS and OS/390 systems 507

- class descriptor table (CDT) *(continued)*
 - supplied classes for z/VM and VM systems 513
- class name
 - activating or deactivating general resource class 444
 - changing general resource profile 307
 - deleting general resource profile 373
 - displaying general resource profile 387
 - list of supplied general resource classes 507, 513
 - specifying
 - for general resource profile 341
- CLASS operand
 - PERMIT command 251
 - SEARCH command 412
- CLASSACT operand
 - SETROPTS command 444
- classes
 - activating 435
 - recording statistics 469
- classroom courses, RACF xii
- CLAUTH (class authority) attribute
 - for existing user 124
 - for new user 55
- CLAUTH operand
 - ADDUSER command 55
 - ALTUSER command 124
- CLIST data set
 - creating 413
- CLIST operand
 - SEARCH command 413
- CMDSYS suboperand
 - ADDUSER command 71
 - ALTUSER command 147
- CMDVIOL operand
 - SETROPTS command 445
- command response logging
 - for new user profile 72
 - for user profile 148, 149
- COMMAND suboperand
 - ADDUSER command 80
 - ALTUSER command 161
- command usage
 - logging for a user 166
- commands
 - summary 1
- COMPATMODE operand
 - SETROPTS command 445
- conditional access list 247
- CONNECT command
 - description 173
 - examples 179
 - RACF requirements 173
 - syntax 174
- connect group
 - ALTUSER command 130
 - CONNECT command 176
- CONNECT group authority
 - description 13
- CONNECT operand
 - RACDCERT command 279

- connect profile
 - assigning group-related attributes 173
 - changing 115, 173
 - creating 48, 173
 - deleting 189
 - displaying with group profile 214
 - displaying with user profile 223
- CONSNAM suboperand
 - ADDUSER command 63
 - ALTUSER command 136
- CONSOLE class
 - description 507
- console command system
 - for new user profile 71
 - for user profile 147
- console message format
 - for new user profile 72
 - for user profile 149
- console message storage
 - for new user profile 73
 - for user profile 151
- console operator command authority
 - for new user 70
- console search key
 - for new user profile 71
 - for user profile 148
- controlled program
 - defining 308, 343
 - specifying access for 254
- controlling
 - access to system for existing user 166
 - access to system for new user 82
 - access to system for terminal 333, 366
- copying access lists 252
- courses about RACF xii
- CPSMOBJ class
 - description 510
- CPSMXMP class
 - description 510
- CPUTIMEMAX operand
 - ADDUSER command 66
- CPUTIMEMAX suboperand
 - ALTUSER command 141
- CREATE group authority
 - description 13
- creating
 - CLIST data set 413
 - model profile 41
- CRITERIA operand
 - RACDCERT command 284
- critical extensions
 - RACDCERT command 264
- CSFKEYS class
 - description 507
- CSFSERV class
 - description 507
- CTL suboperand
 - ADDUSER command 63
 - ALTUSER command 136
- current password
 - changing 244

current RACF options
displaying 454, 456

D

DASD data set
displaying volume information for 417
erase-on-scratch processing
activating 446
deactivating system-wide 447
for existing data set 96
for new data set 39
searching a volume for 417
DASDVOL class
description 507
data application for DFP
changing
for group profile 107
for user profile 128
defining
for new group profile 26
for new user profile 58
data application for LNOTES
changing
for user profile 134
data application for NDS
changing
for user profile 135
data class for DFP
changing
in group profile 107
in user profile 128
defining
for new group profile 26
for new user profile 58
DATA operand
ADDGROUP command 26
ADDSD command 39
ADDUSER command 55
ALTDSD command 95
ALTGROUP command 107
ALTUSER command 125
RALTER command 314
RDEFINE command 352
data set
creating CLIST data set 413
default TSO prefix 17
DFP-managed data set
displaying owner 206
specifying owner 39, 96
logging real data set names 467
protecting single-qualifier named data sets 463
data set profile
changing 90
defining 496
deleting 181
determining RACF protection 496
displaying 200
generic profile 41, 96
model profile
defining 41

data set profile *(continued)*
model profile *(continued)*
model for group data sets 27, 109
using existing profile as model 40
OMVS profile
OMVS for group data sets 28
OVM profile
OVM for group data sets 29
permitting access to 247
searching for
all profiles 411
based on last reference 411
selected profiles 414
tape data set profile 41
DATAAPPL suboperand
ADDGROUP command 26
ADDUSER command 58
ALTGROUP command 107
ALTUSER command 128
database
deactivating or reactivating RACF 399
DATACLAS sub-operand
TARGET command 490
DATACLAS suboperand
ADDGROUP command 26
ADDUSER command 58
ALTGROUP command 107
ALTUSER command 128
DATAENCRYPT operand
RACDCERT command 275
DATASET class
auditing for 442
defining fully-qualified generic profile 496
generic profile checking 448
generic profile command processing 447
global access checking 451
recording statistics for 470
DATASET operand
LISTDSD command 205
RVARY command 404
DATASHARE operand
RVARY command 403
date
syntax 11
days of week
existing user can access system 167
new user can access system 82
terminal can access system 333, 366
DAYS operand
ADDUSER command 82
ALTUSER command 166
RALTER command 333
RDEFINE command 366
DB2
general resource classes 510
DCE
general resource classes 511
DCE operand
LISTUSER command 228
DCICSDCT class
description 510

- deactivating
 - general resource classes 444
 - RACF resource protection using RVARY
 - command 399
 - unused user ID 453
- DEBUG operand
 - RACDCERT command 287
- default group
 - for existing user 128
 - for new user 57
- DEFAULT operand
 - RACDCERT command 280
- DEFTKTLFE operand
 - RALTER command 317
- DELCATEGORY operand
 - ALTDSD command 94
 - ALTUSER command 122
 - RALTER command 309
- DELDIR 1
- DELDOMAINS suboperand
 - ALTUSER command 137
- DELSD command
 - description 181
 - examples 185
 - RACF requirements 182
 - syntax 182
- DELETE operand
 - PERMIT command 251
 - RACDCERT command 270
 - TARGET command 482
- deleting
 - access lists from profile 253
 - console command system
 - from user profile 147
 - console search key
 - from user profile 148
 - data set profile 181
 - general resource profile 371
 - group profile 186
 - message level from user profile 148
 - names from access list 251
 - operator command authority from user profile 147
 - primary language 134
 - secondary language 134
 - security category
 - from data set profile 94
 - from general resource profile 309
 - security level
 - from data set profile 100
 - from general resource profile 324
 - from user profile 160
 - user profile 189
 - volume
 - from tape volume profile 311
- Deleting the distinguished name used by the LDAP proxy server
 - ALTUSER command 157
 - RALTER command 322
- Deleting the LDAP proxy server information
 - ALTUSER command 157
 - RALTER command 323
- Deleting the password used by the LDAP proxy server
 - ALTUSER command 157
 - RALTER command 323
- Deleting the URL of the LDAP proxy server
 - ALTUSER command 156
 - RALTER command 322
- DELFILE 1
- DELGROUP command
 - description 187
 - example 188
 - RACF requirements 186
 - syntax 187
- DELMAP operand
 - RACDCERT command 286
- DELMEM operand
 - RALTER command 311
- DELMSCOPE suboperand
 - ALTUSER command 150
- DELOPCCLASS suboperand
 - ALTUSER command 123, 138
- DELRING operand
 - RACDCERT command 281
- DELUSER command
 - description 189
 - example 192
 - RACF requirements 190
 - syntax 190
- DELVOL operand
 - ALTDSD command 94
 - RALTER command 311
- DESCRIPTION operand
 - TARGET command 483
- DEST suboperand
 - ADDUSER command 80
 - ALTUSER command 162
- destination of SYSOUT data set
 - for existing user 162
 - for new user 80
- DEVICES class
 - description 507
- DFLTGRP operand
 - ADDUSER command 57
 - ALTUSER command 128
- DFP operand
 - ADDGROUP command 26
 - ADDSD command 39
 - ADDUSER command 58
 - ALTDSD command 96
 - ALTGROUP command 107
 - ALTUSER command 128
 - LISTSD command 206
 - LISTGRP command 217
 - LISTUSER command 228
- DFP segment
 - changing
 - in group profile 107
 - in user profile 128
 - defining
 - for new group profile 26
 - for new user profile 58

- DFP segment (*continued*)
 - displaying
 - for group profile 217
 - for user profile 228
- DFP-managed data set
 - displaying owner 206
 - specifying owner 39, 96
- DFSMSdfp
 - general resource classes 512
- DIGTCERT class
 - description 507
- DIGTCRIT class
 - description 507
- DIGTNMAP class
 - description 507
- DIGTRING class
 - description 507
- DIMS class
 - description 511
- DIRACC class
 - description 513
- DIRAUTH class
 - description 508
- DIRECTRY class
 - description 513
- DIRSRCH class
 - description 513
- disability 517
- discrete profile
 - choosing between discrete and generic profiles 500
- data set
 - defining 496, 497
 - deleting 183
 - displaying 207, 389
- general resource
 - defining 342, 501
 - deleting 373
 - displaying 389
- naming 495
- searching for 411
- DISPLAY command
 - description 193
 - examples 195
 - RACF requirements 193
 - syntax 194
- displaying
 - current RACF options 454, 456
 - data set profile 200
 - general resource profile 382
 - group profile 214
 - user profile 223
- DLF object
 - listing
 - general resource profiles 389
 - retain after use 352
 - specifying which can be accessed 352
- DLFCLASS class
 - description 508
- DLFDATA object
 - retaining after use 314
- DLFDATA operand
 - RALTER command 314
 - RDEFINE command 352
 - RLIST command 389
- DLFDATA segment
 - authority to define 339
 - defining 352
- DOCSIGN operand
 - RACDCERT command 275
- documents, licensed xiii
- DOM request reception
 - deleting from user profile 148
 - for new user profile 71
 - for user profile 147
- DOM suboperand
 - ADDUSER command 71
 - ALTUSER command 147
- DOMAIN operand
 - RACDCERT command 276
- DOMAINS suboperand
 - ADDUSER command 63
 - ALTUSER command 137
- DORMANT operand
 - TARGET command 483
- DSN operand
 - RACDCERT command 277
- DSNADM class
 - description 510
- DSNR class
 - description 510

E

- ECICSDCT class
 - description 510
- EGN operand
 - SETROPTS command 445
- EIM (Enterprise Identity Mapping)
 - for existing user 129
 - for new user 59
- EIM domain
 - altering resource options 315
 - options 352
- EIM operand
 - ADDUSER command 59
 - ALTUSER command 129
 - LISTUSER command 228
- EIM segment
 - displaying
 - for user profile 228
- EJBROLE class
 - description 511
- EMAIL operand
 - RACDCERT command 276
- ENCRYPT operand
 - ADDUSER command 59, 317
 - ALTUSER command 131
 - RDEFINE command 354
- enhanced generic naming
 - activating or deactivating 445
 - for data set profile 498

- Enterprise Java Beans
 - general resource classes 511
- ERASE operand
 - ADDSD command 39
 - ALTDSD command 96
 - SETOPTS command 446
- erase-on-scratch processing
 - activating 446
 - deactivating 447
 - for existing DASD data set 96
 - for new DASD data set 39
- event display information
 - for new user profile 72
 - for user profile 150
- exit routine
 - RACF commands that provide 12
- EXPIRED suboperand
 - ALTUSER command 130
- EXPIRES operand
 - SEARCH command 412
- EXPORT operand
 - RACDCERT command 277
- extension logic
 - authorityKeyIdentifier extension 271
 - basicConstraints extension 271
 - issuerAltName extension 271
 - keyUsage extension 271
 - subjectAltName extension 271
 - subjectKeyIdentifier extension 271

F

- FACILITY class
 - description 508, 514
- FCICSFCT class
 - description 510
- FCLASS operand
 - ADDSD command 39
 - PERMIT command 251
 - RDEFINE command 353
- FGENERIC operand
 - ADDSD command 39
 - PERMIT command 252
 - RDEFINE command 353
- FIELD class
 - description 508, 514
- field-level access checking 343
- FILE class
 - description 514
- file sequence number for tape data set 40
- FILEPROCMAX operand
 - ADDUSER command 67
- FILEPROCMAX suboperand
 - ALTUSER command 142
- FILESEQ operand
 - ADDSD command 40
- FILESIZE sub-operand
 - TARGET command 490
- FILTER operand
 - SEARCH command 414

- FIMS class
 - description 511
- FORMAT operand
 - RACDCERT command 277
- FROM operand
 - ADDSD command 40
 - PERMIT command 252
 - RDEFINE command 353
- FSOBJ class
 - description 513
- FSSEC class
 - description 513
- fully-qualified generic profile
 - naming 496
- FUNCTION operand
 - HELP command 199
- FVOLUME operand
 - ADDSD command 40
 - PERMIT command 252
 - RDEFINE command 354

G

- GCICSTRN class
 - description 510
- GCPSMOBJ class
 - description 510
- GCSFKEYS class
 - description 508
- GDASDVOL class
 - description 508
- GDG (generation data group)
 - activating model profile for 459
- GDSNBP class
 - description 510
- GDSNCL class
 - description 510
- GDSNDB class
 - description 510
- GDSNJR class
 - description 510
- GDSNPK class
 - description 510
- GDSNPN class
 - description 510
- GDSNSC class
 - description 510
- GDSNSG class
 - description 510
- GDSNSM class
 - description 510
- GDSNSP class
 - description 510
- GDSNTB class
 - description 510
- GDSNTS class
 - description 510
- GDSNUF class
 - description 510
- GDSNUT class
 - description 510

- GEJBROLE class
 - description 511
- GENCERT operand
 - RACDCERT command 270
- GENCMD operand
 - SETROPTS command 447
- general resource class 510
 - activating 444
 - auditing for 442
 - deactivating 444
 - generic profile checking 448
 - generic profile command processing 447
 - global access checking 347, 451
 - product use of
 - CICS 509
 - DB2 510
 - DCE 511
 - DFSMSdtp 512
 - Enterprise Java Beans 511
 - IMS 511
 - Infoprint Server 511
 - Information/Management 511
 - LFS/ESA 511
 - License Manager 512
 - Lotus Notes for z/OS 512
 - MQSeries 512
 - NetView 512
 - Novell Directory Services for OS/390 512
 - Security Server Network Authentication Service 512
 - SMS 512
 - Tivoli 512
 - Tivoli Service Desk 511
 - TSO 513
 - z/OS UNIX 513
 - recording statistics for 470
 - supplied 507, 513
- general resource profile
 - changing 303
 - defining 338, 501
 - deleting 371
 - determining RACF protection 496
 - displaying 382
 - permitting access to 247
 - searching for based on last reference 411
 - searching RACF database for 414
 - using existing profile as model for 353
- generic character
 - defining generic profile name 495
 - GENERIC operand in place of 41
- GENERIC operand
 - ADDSD command 41
 - ALTDSD command 96
 - DELDSD command 183
 - LISTDSD command 207
 - PERMIT command 252
 - RLIST command 389
 - SEARCH command 411
 - SETROPTS command 448
- generic profile
 - choosing between discrete and generic profiles 500

- generic profile (*continued*)
 - data set
 - defining, enhanced generic naming active 498
 - defining, enhanced generic naming inactive 497
 - deleting 183
 - displaying 207
 - using existing profile as generic 96
 - using new profile as generic 41
 - displaying for a data set 200
 - general resource
 - defining 342, 501
 - deleting 373
 - displaying 389
 - naming 495
 - refreshing in-storage profiles 467
 - searching for 411
 - generic profile checking 449, 450
 - activating or deactivating 448
 - generic profile command processing
 - activating or deactivating 447
 - GENERICOWNER operand
 - SETROPTS command 449
 - GENLIST operand
 - SETROPTS command 450
 - GENREQ operand
 - RACDCERT command 278
 - GID operand
 - SEARCH command 415
 - GID suboperand
 - ADDGROUP command 29
 - ALTGROUP command 28, 110, 111
 - GIMS class
 - description 511
 - GINFOMAN class
 - description 511
 - global access checking
 - activating or deactivating 451
 - defining entry in table 346
 - refreshing in-storage table 467
 - GLOBAL class
 - description 508, 514
 - GLOBAL operand
 - SETROPTS command 451
 - GLOBALAUDIT operand
 - ALTDSD command 97
 - RALTER command 316
 - GMBR class
 - description 508, 514
 - GMQADMIN class
 - description 512
 - GMQNLIST class
 - description 512
 - GMQPROC class
 - description 512
 - GMQUEUE class
 - description 512
 - group
 - default for existing user 128
 - default for new user 57
 - group-related user attributes
 - assigning for user 173

- group (*continued*)
 - maximum number of users in 175
- group authority
 - description 13
 - for existing user 122
 - for new user 54
 - in user's connect profile 176
- group data set
 - defining 34
 - model profile processing 459
- group name
 - as new owner
 - of data set profiles of removed user 378
 - as owner
 - of connect profile 177
 - of data set profile 98
 - of general resource profile 321
 - of group profile 112
 - of new data set profile 42
 - of new general resource profile 357
 - of new group profile 30
 - of new user profile 75
 - of user profile 154
 - changing access to resource for 252
 - deleting group profile 188
 - displaying
 - data set profiles for 205
 - group profile for 216
 - for existing group 106
 - for new group 26
 - for removing user from group 378
 - syntax 10
- GROUP operand
 - ALTUSER command 130
 - CONNECT command 176
 - DISPLAY command 195
 - RALTER command
 - =MEMBER operand 327
 - RDEFINE command
 - =MEMBER operand 362
 - REMOVE command 378
 - SIGNOFF command 477
- group profile
 - changing 104
 - defining 24
 - deleting 186
 - displaying 214
 - searching for based on last reference 411
- group-AUDITOR attribute
 - in user's connect profile 176
- group-OPERATIONS attribute
 - in user's connect profile 177
 - logging activities for 460
- group-SPECIAL attribute
 - in user's connect profile 179
 - logging activities for 468
- GRPACC (group access) attribute
 - for existing user 130
 - for new user 59
 - in user's connect profile 176

- GRPACC operand
 - ADDUSER command 59
 - ALTUSER command 130
 - CONNECT command 176
- GRPLIST operand
 - SETOPTS command 452
- GSDSF class
 - description 508
- GTERMINL class
 - description 508, 514

H

- HANDSHAKE operand
 - RACDCERT command 275
- HCICSFCT class
 - description 510
- HELP command
 - description 198
 - examples 199
 - syntax 198
- hierarchical storage manager (HSM)
 - TVTOC operand
 - RALTER command 332
- HIGHTRUST operand
 - RACDCERT command 266, 270
- HIMS class
 - description 511
- HISTORY operand
 - LISTDSD command 208
 - RLIST command 390
- HISTORY suboperand
 - PASSWORD operand 460
- hold class for TSO
 - for existing user 162
 - for new user 80
- HOLDCLASS suboperand
 - ADDUSER command 80
 - ALTUSER command 162
- HOME suboperand
 - ADDUSER command 67
 - ALTUSER command 142

I

- IC suboperand
 - ADDUSER command 64
 - ALTUSER command 137
- ICSF operand
 - RACDCERT command 268, 275
- ID operand
 - LISTDSD command 205
 - PERMIT command 252
 - RACDCERT command 261
- IDNFILTER operand
 - RACDCERT command 282
- ILMADMIN class
 - description 512
- IMS (Information Management System)
 - general resource classes 511

- in-storage profile
 - SETROPTS GENLIST processing for 450
 - SETROPTS RACLIST processing for 464
- INACTIVE operand
 - RVARY command 402
 - SETROPTS command 453
- INCLUDE operand
 - SET command 426
- INFOMAN class
 - description 511
- Infoprint Server
 - general resource class 511
- Information/Management
 - general resource classes 511
- INITSTATS operand
 - SETROPTS command 453
- installation exit routine
 - RACF commands that provide 12
- installation-defined data
 - changing
 - in data set profile 95
 - in general resource profile 314
 - in group profile 107
 - in user profile 125
 - defining
 - for data set profile 39
 - for general resource profile 352
 - for group profile 26
 - for new user profile 55
 - deleting
 - from general resource profile 314
 - displaying
 - for general resource profile 383
 - from data set profile 200
 - from group profile 214
 - user profile 223
- INTERVAL operand
 - PASSWORD command 244
- INTERVAL suboperand
 - PASSWORD operand 460
- IP operand
 - RACDCERT command 276
- IPCOBJ class
 - description 513
- ISPF panels
 - compared to RACF TSO commands 16
 - not affected by SETROPTS LANGUAGE setting 455
 - sample for password rules 17

J

- JAVA class
 - description 511
- JCICSJCT class
 - description 510
- JES (job entry subsystem)
 - activating or deactivating options for 453
- JES operand
 - SETROPTS command 453

- JESINPUT class
 - description 508
- JESJOBS class
 - description 508
- JESSPOOL class
 - description 508
- job class for TSO
 - for existing user 162
 - for new user 80
- job entry subsystem (JES)
 - See JES (job entry subsystem)
- JOBCLASS suboperand
 - ADDUSER command 80
 - ALTUSER command 162
- JOBNAMEs suboperand
 - DLFDATA operand 352
- JOIN group authority
 - description 13

K

- KCICSJCT class
 - description 510
- KERB operand
 - ADDUSER command 59
 - ALTUSER command 131
 - LISTUSER command 228
 - RALTER command 316
 - RDEFINE command 354
 - RLIST command 390
- KERB segment
 - displaying
 - for user profile 228
- KERBLINK class
 - description 512
- KERBLVL operand
 - SETROPTS command 454
- KERBNAME operand
 - ADDUSER command 60
 - ALTUSER command 132
 - RALTER command 318
 - RDEFINE command 355
- key
 - See application key
- KEY suboperand
 - ADDUSER command 71
 - ALTUSER command 148
- keyboard 517
- KEYENCRYPTED suboperand
 - RALTER command 326
 - RDEFINE command 362
- KEYMASKED suboperand
 - RALTER command 326
 - RDEFINE command 361
- KeyUsage operand
 - RACDCERT command 275

L

- LABEL operand
 - RACDCERT command 281

- LAN File Services/ESA (LFS/ESA)
 - See LFS/ESA (LAN File Services/ESA)
- LANGUAGE operand
 - ADDUSER command 61
 - ALTUSER command 133
 - LISTUSER command 228
- LANGUAGE segment
 - alter primary language 134
 - alter secondary language 134
 - delete primary language 134
 - delete secondary language 134
 - NOPRIMARY suboperand 134
 - NOSECONDARY suboperand 134
 - PRIMARY suboperand 134
 - SECONDARY suboperand 134
 - user profile
 - displaying 228
- last reference
 - searching for profile based on 411
- LDAPHOST suboperand
 - ADDUSER command 76
 - ALTUSER command 155
 - RALTER command 321
 - RDEFINE command 357
- LDIRECT 1
- LENGTH suboperand
 - RULEn suboperand 461
- level indicator
 - as search criteria 412
 - changing
 - for data set profile 98
 - defining
 - for data set profile 41
 - for general resource profile 320, 356
- LEVEL operand
 - ADDSD command 41
 - ALTDSD command 98
 - RALTER command 320
 - RDEFINE command 356
 - SEARCH command 412
- LEVEL suboperand
 - ADDUSER command 71
 - ALTUSER command 148
- LFILE 1
- LFS/ESA (LAN File Services/ESA)
 - general resource class 511
- LFSCCLASS class
 - description 512
- library name
 - for controlled program 349
- License Manager
 - general resource class 512
- licensed documents xiii
- limiting
 - access
 - to a terminal 333, 366
 - to system for existing user 166
 - to system for new user 82
- LIST operand
 - RVARY command 404
 - SEARCH command 415
- LIST operand (*continued*)
 - SET command 427
 - SETOPTS command 456
 - TARGET command 484
- list-of-groups checking
 - activating or deactivating 452
- LISTDSD command
 - description 200
 - examples 209
 - RACF requirements 202
 - syntax 204
- LISTGRP command
 - description 214
 - examples 218
 - RACF requirements 215
 - syntax 216
- LISTMAP operand
 - RACDCERT command 286
- LISTRING operand
 - RACDCERT command 281
- LISTUSER command
 - description 223
 - examples 231
 - RACF requirements 224
 - syntax 226
- LNOTES operand
 - ADDUSER command 62
 - ALTUSER command 134
 - LISTUSER command 229
- LNOTES segment
 - changing
 - in user profile 134
 - user profile
 - displaying 229
- LOCAL operand
 - TARGET command 485
- locating profiles in RACF database 408
- LOGCMDRESP suboperand
 - ADDUSER command 72
 - ALTUSER command 148
- logging
 - access attempt
 - for existing general resource profile 313
 - for new general resource profile 351
 - access attempts
 - based on security level 469
 - for existing data set profile 95
 - for new data set profile 38
 - activities for OPERATIONS attribute 460
 - activities for SPECIAL attribute 468
 - RACF command usage by a user 166
 - real data set names 467
 - system-wide command violations 445
 - system-wide for RACF classes 442
- logon procedure for TSO
 - changing 163
 - defining 81
- LookAt message retrieval tool xiii
- Lotus Notes for z/OS
 - general resource class 512

M

- MAIN operand
 - TARGET command 485
- management class for DFP
 - changing
 - for group profile 108
 - in user profile 129
 - defining
 - for group profile 27
 - for user profile 58
- MAP operand
 - RACDCERT command 281
- MASK operand
 - SEARCH command 415
- maximum number of users in group 175
- maximum TSO region size
 - for existing user 162
 - for new user 80
- MAXSIZE suboperand
 - ADDUSER command 80
 - ALTUSER command 162
- MAXTKTLFE operand
 - ADDUSER command 61
 - ALTUSER command 133
 - RALTER command 318
 - RDEFINE command 356
- MCICSPPT class
 - description 510
- MDSNBP class
 - description 510
- MDSNCL class
 - description 510
- MDSNDB class
 - description 510
- MDSNJR class
 - description 511
- MDSNPK class
 - description 511
- MDSNPN class
 - description 511
- MDSNSC class
 - description 511
- MDSNSG class
 - description 511
- MDSNSM class
 - description 511
- MDSNSP class
 - description 511
- MDSNTB class
 - description 511
- MDSNTS class
 - description 511
- MDSNUF class
 - description 511
- MDSNUT class
 - description 511
- member
 - adding to resource group 309, 344
 - deleting from resource group 311
- message
 - notify when profile denies access
 - for existing data set profile 98
 - for existing general resource profile 320
 - for new data set profile 42
 - for new general resource profile 357
 - password expiration 462
 - warning
 - changing for data set profile 101
 - defining for data set profile 45
 - for existing general resource profile 332
 - for new general resource profile 366
- message class for TSO
 - changing for user 163
 - defining for user 81
- message retrieval tool, LookAt xiii
- message routing codes
 - for new user profile 73
 - for user profile 151
- MFORM suboperand
 - ADDUSER command 72
 - ALTUSER command 149
- MGMTCLAS class
 - description 512
- MGMTCLAS sub-operand
 - TARGET command 490
- MGMTCLAS suboperand
 - ADDGROUP command 27
 - ADDUSER command 58
 - ALTGROUP command 108
 - ALTUSER command 129
- MIGID suboperand
 - ADDUSER command 72
 - ALTUSER command 149
- migration id assignment
 - for user profile 149
- migration ID assignment
 - for new user profile 72
- minimum TSO region size
 - for existing user 164
 - for new user 81
- MINTKTLFE operand
 - RALTER command 319
 - RDEFINE command 356
- MMAPAREAMAX operand
 - ADDUSER command 68
- MMAPAREAMAX suboperand
 - ALTUSER command 143
- MMSLSTxx PARMLIB member
 - relation to SETROPTS LANGUAGE setting 455
- model data set profile
 - authorization required to specify 35
 - copying fields from 35
 - defining 41
 - displaying name 214
 - for existing user 135
 - for group data sets 109
 - for new user 62
 - locating volume for 40
 - model for group data sets 27
 - searching for 411

- model data set profile *(continued)*
 - system-wide processing options 459
 - using existing profile as model 40
- model general resource profile
 - using existing profile as 353
 - using volume to locate 354
- MODEL operand
 - ADDGROUP command 27
 - ADDSD command 41
 - ADDUSER command 62
 - ALTGROUP command 109
 - ALTUSER command 135
 - SEARCH command 411
 - SETROPTS command 459
- MONITOR suboperand
 - ADDUSER command 72
 - ALTUSER command 150
- MQADMIN class
 - description 512
- MQCMDS class
 - description 512
- MQCONN class
 - description 512
- MQNLIST class
 - description 512
- MQPROC class
 - description 512
- MQQUEUE class
 - description 512
- MQSeries
 - general resource classes 512
- MSCOPE suboperand
 - ADDUSER command 73
 - ALTUSER command 150
- MSGCLASS suboperand
 - ADDUSER command 81
 - ALTUSER command 163
- MSGRECV suboperand
 - ADDUSER command 64
 - ALTUSER command 138
- MULTIID operand
 - RACDCERT command 261

N

- NAME operand
 - ADDUSER command 62
 - ALTUSER command 135
- NCICSPPT class
 - description 510
- NDS operand
 - ADDUSER command 62
 - ALTUSER command 135
 - LISTUSER command 229
- NDS segment
 - changing
 - in user profile 135
 - user profile
 - displaying 229
- NDSLINK class
 - description 512

- NETCMDS class
 - description 512
- NETSPAN class
 - description 512
- NetView
 - general resource classes 512
- NETVIEW operand
 - ADDUSER command 63
 - ALTUSER command 136
 - LISTUSER command 229
- NETVIEW segment
 - ADDDOMAINS suboperand 137
 - ADDOPCLASS suboperand 138
 - changing
 - in user profile 136
 - CONSNAME suboperand 136
 - CTL suboperand 136
 - DELDOMAINS suboperand 137
 - DELOPCLASS suboperand 138
 - DOMAINS suboperand 137
 - group profile
 - displaying 229
 - IC suboperand 137
 - MSGRECV suboperand 138
 - NGMFADMN suboperand 138
 - NOCONSNAME suboperand 136
 - NOCTL suboperand 137
 - NODOMAINS suboperand 137
 - NOIC suboperand 137
 - NOMSGRECV suboperand 138
 - NONGMFADMN suboperand 138
 - NOOPCLASS suboperand 138
 - OPCLASS suboperand 138
- Network Authentication Service
 - general resource classes 512
- new group
 - defining 24
- new password
 - specifying 244
- new user
 - defining 48
- NEWLABEL operand
 - RACDCERT command 270, 286
- NGMFADMN suboperand
 - ADDUSER command 64
 - ALTUSER command 138
- NGMFVSPN suboperand
 - ADDUSER command 64
 - ALTUSER command 138
- no security label for TSO
 - for existing user 164
- NOACCTNUM suboperand
 - ALTUSER command 161
- NOADDCREATOR operand
 - SETROPTS command 441
- NOADSP operand
 - ADDUSER command 53
 - ALTUSER command 122
 - CONNECT command 175
 - SETROPTS command 441

NOAPPLAUDIT operand
 SETROPTS command 442
 NOAPPLDATA operand
 RALTER command 313
 NOASSIZEMAX suboperand
 ALTUSER command 139
 NOAUDIT operand
 SETROPTS command 442
 NOAUDITOR operand
 ADDUSER command 54
 ALTUSER command 122
 CONNECT command 176
 NOAUTH suboperand
 ALTUSER command 147
 NOAUTO suboperand
 ALTUSER command 147
 NOAUTODIRECT operand
 SET command 425
 NOBINDDN suboperand
 ALTUSER command 157
 RALTER command 322
 NOBINDPW suboperand
 ALTUSER command 157
 RALTER command 323
 NOCICS operand
 ALTGROUP command 124
 NOCLASSACT operand
 RVARY command 402
 SETROPTS command 444
 NOCLAUTH operand
 ADDUSER command 55
 ALTUSER command 125
 NOCMDSYS suboperand
 ALTUSER command 147
 NOCMDVIOL operand
 SETROPTS command 445
 NOCOMPATMODE operand
 SETROPTS command 445
 NOCONSNAME suboperand
 ALTUSER command 136
 NOCPUTIMEMAX suboperand
 ALTUSER command 142
 NOCTL suboperand
 ALTUSER command 137
 NODATA operand
 ALTDSD command 96
 ALTGROUP command 107
 ALTUSER command 125
 RALTER command 314
 NODATAAPPL suboperand
 ALTGROUP command 107
 ALTUSER command 128
 NODATACLAS suboperand
 ALTGROUP command 108
 ALTUSER command 128
 NODATASHARE operand
 RVARY command 403
 NODE operand
 TARGET command 485
 NODEFTKTLFE operand
 RALTER command 317
 NODES class
 description 508
 NODEST suboperand
 ALTUSER command 162
 NODFP operand
 ALTGROUP command 109
 ALTUSER command 129
 NODMBR class
 description 508
 NODOM suboperand
 ALTUSER command 148
 NODOMAINS suboperand
 ALTUSER command 137
 NOEGN operand
 SETROPTS command 446
 NOENCRYPT operand
 ALTUSER command 131, 317
 NOERASE operand
 ALTDSD command 96
 SETROPTS command 447
 NOEXPIRED suboperand
 ALTUSER command 130
 NOFILEPROCMAx suboperand
 ALTUSER command 142
 NOGENCMD operand
 SETROPTS command 448
 NOGENERIC operand
 LISTDSD command 207
 RLIST command 389
 SEARCH command 411
 SETROPTS command 449
 NOGENERICOWNER operand
 SETROPTS command 450
 NOGENLIST operand
 SETROPTS command 451
 NOGID suboperand
 ALTGROUP command 111
 NOGLOBAL operand
 SETROPTS command 452
 NOGROUP operand
 RALTER command 327
 NOGRPACC operand
 ADDUSER command 59
 ALTUSER command 131
 CONNECT command 176
 NOGRPLIST operand
 SETROPTS command 452
 NOHISTORY suboperand
 PASSWORD operand 460
 NOHOLDCLASS suboperand
 ALTUSER command 162
 NOHOME suboperand
 ALTUSER command 143
 NOIC suboperand
 ALTUSER command 137
 NOINACTIVE operand
 SETROPTS command 453
 NOINITSTATS operand
 SETROPTS command 453
 NOINTERVAL operand
 PASSWORD command 244

NOJOBCLASS suboperand	
ALTUSER command	162
NOKERB operand	
ALTUSER command	133
NOKERBNAME operand	
ALTUSER command	133
RALTER command	318
NOKEY suboperand	
ALTUSER command	148
NOLANGUAGE operand	
ALTUSER command	134
NOLDAPHOST suboperand	
ALTUSER command	156
RALTER command	322
NOLEVEL suboperand	
ALTUSER command	148
NOLIST operand	
RVARY command	404
SEARCH command	415
NOLOGCMDRESP suboperand	
ALTUSER command	149
NOMASK operand	
SEARCH command	416
NOMAXSIZE suboperand	
ALTUSER command	163
NOMAXTKLFE operand	
ALTUSER command	133
RALTER command	319
NOMFORM suboperand	
ALTUSER command	149
NOMGMTCLAS suboperand	
ALTGROUP command	108
ALTUSER command	129
NOMIGID suboperand	
ALTUSER command	149
NOMINTKTLFE operand	
RALTER command	319
NOMMAPAREAMAX suboperand	
ALTUSER command	143
NOMODEL operand	
ALTGROUP command	109
ALTUSER command	135
SETROPTS command	460
NOMONITOR suboperand	
ALTUSER command	150
NOMSCOPE suboperand	
ALTUSER command	151
NOMSGCLASS suboperand	
ALTUSER command	163
NOMSGRECV suboperand	
ALTUSER command	138
nonautomatic TAPEVOL profile	
altering	303
defining	365
permitting access to	248
NONGMFADMN suboperand	
ALTUSER command	138
NONGMFVSPN suboperand	
ALTUSER command	138
NONOTIFY operand	
ALTDSD command	98
NONOTIFY operand	<i>(continued)</i>
RALTER command	321
NONOTIFY sub-operand	
SET command	425
NONVSAM operand	
SEARCH command	411
NOOIDCARD operand	
ADDUSER command	64
ALTUSER command	139
NOOMVS operand	
ALTGROUP command	111
NOOMVS suboperand	
ALTUSER command	145
NOOPCLASS suboperand	
ALTUSER command	123, 138
NOOPERATIONS operand	
ADDUSER command	69
ALTUSER command	145
CONNECT command	177
NOOPERAUDIT operand	
SETROPTS command	460
NOOPERPARM operand	
ALTUSER command	152
NOOUTPUT sub-operand	
SET command	425
NOOVM operand	
ALTGROUP command	112
NOPADCHK suboperand	
RDEFINE command	349
NOPASSWORD operand	
ADDUSER command	75
ALTUSER command	155
RALTER command	320
NOPREFIX operand	
SETROPTS command	463
NOPRIMARY suboperand	
ALTUSER command	134
NOPRIVILEGED operand	
RALTER command	327
NOPROC suboperand	
ALTUSER command	164
NOPROCUSERMAX suboperand	
ALTUSER command	144
NOPROGRAM suboperand	
ALTUSER command	145
NOPROTECTALL operand	
SETROPTS command	464
NOPROXY operand	
ALTUSER command	157
RALTER command	323
NORACF operand	
LISTGRP command	217
LISTUSER command	229
NORACLIST operand	
SETROPTS command	466
NOREALDSN operand	
SETROPTS command	467
NORESTRICTED operand	
ALTUSER command	78, 158
NOREVOKE suboperand	
PASSWORD operand	461

NOROUTCODE suboperand
 ALTUSER command 151
 NORULEn suboperand
 PASSWORD operand 462
 NORULES suboperand
 PASSWORD operand 462
 NOSAUDIT operand
 SETROPTS command 468
 NOSECLABEL operand
 ALTDSD command 99
 ALTUSER command 159
 RALTER command 323
 NOSECLABEL suboperand
 ALTUSER command 164
 NOSECLABELAUDIT operand
 SETROPTS command 468
 NOSECLEVEL operand
 ALTDSD command 100, 113
 ALTUSER command 160
 RALTER command 324
 NOSECLEVELAUDIT operand
 SETROPTS command 469
 NOSECONDARY suboperand
 ALTUSER command 134
 NOSET operand
 ADDSD command 41
 ALTDSD command 97
 DELDSD command 184
 NOSINGLEDSN operand
 RALTER command 325
 NOSIZE suboperand
 ALTUSER command 164
 NOSNAME suboperand
 ALTUSER command 135
 NOSPECIAL operand
 ADDUSER command 79
 ALTUSER command 160
 CONNECT command 179
 NOSTATISTICS operand
 SETROPTS command 470
 NOSTDATA operand
 RALTER command 326, 328
 NOSTORAGE suboperand
 ALTUSER command 151
 NOSTORCLAS suboperand
 ALTGROUP command 109
 ALTUSER command 129
 NOSYSOUTCLASS suboperand
 ALTUSER command 165
 NOTAFTERDATE operand
 RACDCERT command 274
 NOTAPE suboperand
 RVARY command 403
 NOTAPEDSN operand
 SETROPTS command 470
 NOTBEFOREDATE operand
 RACDCERT command 274
 NOTELINK class
 description 512
 NOTERMUACC operand
 ADDGROUP command 30
 NOTERMUACC operand *(continued)*
 ALTGROUP command 113
 NOTHEADSMAX suboperand
 ALTUSER command 144
 NOTIFY operand
 ADDSD command 42
 ALTDSD command 98
 RALTER command 320
 RDEFINE command 357
 NOTIFY sub-operand
 SET command 423
 NOTIMEOUT suboperand
 ALTUSER command 124
 NOTIMEZONE operand
 RALTER command 329
 NOTRACE operand
 RALTER command 327
 NOTRUST operand
 RACDCERT command 266, 270, 286
 NOTRUSTED operand
 RALTER command 328
 NOTSO operand
 ALTUSER command 165
 NOTVTOC operand
 RALTER command 332
 NOUAUDIT operand
 ALTUSER command 166
 NOUD suboperand
 ALTUSER command 152
 NOUID suboperand
 ALTUSER command 141
 NOUNAME suboperand
 ALTUSER command 136
 NOUNIT suboperand
 ALTUSER command 165
 NOUSER operand
 RALTER command 327
 NOUSERDATA suboperand
 ALTUSER command 165
 Novell Directory Services for OS/390
 general resource class 512
 NOWARNING operand
 ALTDSD command 101
 RALTER command 333
 NOWARNING suboperand
 PASSWORD operand 462
 NOWHEN(PROGRAM) operand
 SETROPTS command 471
 NOYOURACC operand
 RLIST command 391
 NVASAPDT class
 description 512

O

OIDCARD (operator identification card)
 See operator identification card (OIDCARD)
 OIDCARD operand
 ADDUSER command 64
 ALTUSER command 138

- OIMS class
 - description 511
- OMVS operand
 - ADDGROUP command 28
 - ADDUSER command 65
 - ALTGROUP command 109
 - ALTUSER command 139
 - LISTGRP command 217
 - LISTUSER command 229
- OMVS profile
 - OMVS for group data sets 28
- OMVS segment
 - changing
 - in group profile 109
 - in user profile 139
 - defining
 - for new user profile 65
 - group profile
 - displaying 217
 - user profile
 - displaying 229
- ONLYAT operand
 - ADDGROUP command 26
 - ADDSD command 38
 - ADDUSER command 53
 - ALTDSD command 95
 - ALTGROUP command 107
 - ALTUSER command 122
 - CONNECT command 175
 - DELDSD command 183
 - DELGROUP command 188
 - DELUSER command 191
 - LISTDSD command 205
 - LISTGRP command 217
 - LISTUSER command 227
 - PASSWORD command 244
 - PERMIT command 251
 - RALTER command 313
 - RDEFINE command 351
 - RDELETE command 374
 - REMOVE command 378
 - RLIST command 388
 - SEARCH command 412
 - SETROPTS command 442
- OPCLASS suboperand
 - ADDUSER command 64
 - ALTUSER command 123, 138
- OPERANDS operand
 - HELP command 199
- OPERATIONS attribute
 - for existing user 145
 - for new user 69
 - logging activities for 460
- OPERATIONS operand
 - ADDUSER command 69
 - ALTUSER command 145
 - CONNECT command 177
- OPERATIVE operand
 - TARGET command 483
- operator identification card (OIDCARD)
 - for existing user 138
 - operator identification card (OIDCARD) (*continued*)
 - for new user 64
- operator information
 - changing command authority 146
 - delete from profile 152
 - for user profile 146
- OPERAUDIT operand
 - SETROPTS command 460
- OPERCMD class
 - description 508
- OPERPARM operand
 - ADDUSER command 70
 - ALTUSER command 146
 - LISTUSER command 230
- OPERPARM segment
 - alter operator command authority
 - for user profile 146
 - AUTH suboperand 146
 - AUTO request reception
 - delete from user profile 147
 - AUTO suboperand 70, 147
 - CMDSYS suboperand 71, 147
 - command response logging
 - for new user profile 72
 - for user profile 148, 149
 - console command system
 - for new user profile 71
 - for user profile 147
 - console message format
 - for new user profile 72
 - for user profile 149
 - console message storage
 - for new user profile 73
 - for user profile 151
 - console operator command authority
 - for new user 70
 - console search key
 - for new user profile 71
 - for user profile 148
 - deleting
 - console command system from user profile 147
 - console search key from user profile 148
 - from profile 152
 - message level from user profile 148
 - operator command authority from user profile 147
 - displaying for user profile 230
- DOM request reception
 - deleting from user profile 148
 - for new user profile 71
 - for user profile 147
- DOM suboperand 71, 147
- event display information
 - for new user profile 72
 - for user profile 150
- KEY suboperand 71, 148
- LEVEL suboperand 71, 148
- LOGCMDRESP suboperand 72, 148
- message routing codes
 - for new user profile 73
 - for user profile 151

OPERPARM segment *(continued)*
 MFORM suboperand 72, 149
 MIGID suboperand 72, 149
 migration id assignment
 for user profile 149
 migration ID assignment
 for new user profile 72
 MONITOR suboperand 72, 150
 MSCOPE suboperand 73, 150
 NOAUTH suboperand 147
 NOAUTO suboperand 147
 NOCMDSYS suboperand 147
 NODOM suboperand 148
 NOKEY suboperand 148
 NOLEVEL suboperand 148
 NOLOGCMDRESP suboperand 149
 NOMFORM suboperand 149
 NOMIGID suboperand 149
 NOMONITOR suboperand 150
 NOMSCOPE suboperand 151
 NOROUTCODE suboperand 151
 NOSTORAGE suboperand 151
 NOUD suboperand 152
 ROUTCODE suboperand 73, 151
 STORAGE suboperand 73, 151
 system message reception
 for new user profile 73
 for user profile 150
 type of broadcast messages
 for new user profile 71
 for user profile 148
 UD suboperand 74, 152
 undelivered message reception
 for new user profile 74
 for user profile 152
 user profile
 for new user 70
 options
 displaying current RACF 454, 456
 setting system-wide RACF 435
 OUTPUT sub-operand
 SET command 425
 OVM operand
 ADDGROUP command 29
 ALTGROUP command 111
 LISTUSER command 230
 OVM profile
 OVM for group data sets 29
 OVM segment
 changing
 in group profile 111
 displaying for user profile 230
 OVM suboperand
 ADDUSER command 74
 ALTUSER command 152
 OWNER operand
 ADDGROUP command 30
 ADDSD command 42
 ADDUSER command 75
 ALTDSD command 98
 ALTGROUP command 112

OWNER operand *(continued)*
 ALTUSER command 154
 CONNECT command 177
 RALTER command 321
 RDEFINE command 357
 REMOVE command 378

P

PADCHK suboperand
 RDEFINE command 349
 password
 canceling syntax rules 462
 change interval 244, 460
 changing
 current 244
 number of passwords to be saved 460
 system-wide options 460
 for password-protected data sets 37, 93
 for RVAR command processing 468
 initial logon for new user 75
 logon for existing user 154
 never expires 244
 specifying
 consecutive invalid passwords to revoke user
 ID 461
 new 244
 syntax rules 461
 warning message for password expiration 462
 PASSWORD command
 description 242
 examples 245
 RACF requirements 243
 syntax 243
 PASSWORD JCL statement 18
 PASSWORD operand
 ADDUSER command 75
 ALTUSER command 154
 PASSWORD command 244
 RACDCERT command 268, 278
 RALTER command 319
 RDEFINE command 356
 SETROPTS command 460
 password rules
 ISPF panel for 16
 password-protected data set
 password when altering profile 93
 specifying password for 37
 PCICSPSB class
 description 510
 PERFGRP class
 description 513
 PERMDIR 1
 PERMFILE 1
 PERMIT command
 description 247
 examples 256
 RACF requirements 249
 syntax 249
 permitting access to profiles 249
 persistent verification 193, 360, 476

- PIMS class
 - description 511
- PMBR class
 - description 508
- POE operand
 - DISPLAY command 194
 - SIGNOFF command 477
- PREFIX operand
 - LISTDSD command 205
 - SETOPTS command 463
 - TARGET command 485
- preventing
 - access to profiles 249
 - user from accessing system 158, 178
- PRIMARY suboperand
 - ALTUSER command 134
- Print Services Facility (PSF)
 - See PSF (Print Services Facility)
- PRINTSRV class
 - description 511
- PRIVILEGED operand
 - RALTER command 327
 - RDEFINE command 363
- PROC suboperand
 - ADDUSER command 81
 - ALTUSER command 163
- PROCACT class
 - description 513
- PROCESS class
 - description 513
- PROCUSER operand
 - ADDUSER command 68
- PROCUSERMAX suboperand
 - ALTUSER command 143
- profile
 - defining with enhanced generic naming 498
 - locating in the RACF database 408
- profile name
 - data set profile
 - changing 93
 - defining new 36
 - deleting 183
 - displaying 205
 - using as model 40
 - general resource profile
 - changing 308
 - defining 342
 - deleting 373
 - displaying 387
 - using for model profile 353
 - modifying access list for 250
 - syntax 11
- PROGRAM class
 - description 508
- program control
 - activating 471
 - conditional access list 247, 255
 - controlled program
 - access for 254
 - defining 308, 343
 - deactivating 471
- program control (*continued*)
 - in-storage table
 - changing 348
 - refreshing 467
- PROGRAM suboperand
 - ADDUSER command 68
 - ALTUSER command 144
- PROPCNTL class
 - description 508
- protect-all processing
 - activating or deactivating 463
- PROTECTALL operand
 - SETOPTS command 463
- protected user ID 76, 155
- PROTOCOL operand
 - TARGET command 486
- PROXY operand
 - ADDUSER command 76
 - ALTUSER command 155
 - LISTUSER command 230
 - RALTER command 321
 - RDEFINE command 357
 - RLIST command 391
- PROXY segment
 - displaying for user profile 230
- PSF (Print Services Facility)
 - general resource class 509
- PSFMPL class
 - description 509, 514
- PTKTDATA class
 - description 509, 514
- PTKTVAL class
 - description 512, 514
- publications
 - on CD-ROM xii
 - softcopy xii
- PURGE operand
 - TARGET command 487
- PWSYNC operand
 - SET command 427

Q

- QCICSPSB class
 - description 510
- QIMS class
 - description 511

R

- RACDCERT command
 - description 258
 - examples 289
 - syntax 259
- RACF
 - classroom courses xii
 - publications
 - on CD-ROM xii
 - softcopy xii
- RACF administration
 - classroom courses xii

- RACF commands
 - basic information for issuing RACF Commands 7
 - descriptions 15
 - logging usage by a user 166
 - obtaining help for 198
 - operator commands 21
 - return codes 11
 - summary of 1
 - symbols used in syntax diagrams 11
 - syntax 9
- RACF indication
 - for existing data set profile 97
 - for new data set profile 42
- RACF operator commands
 - descriptions 21
 - entering 21
- RACF options
 - displaying current 454, 456
 - example of display 475
- RACF protection
 - removing from data set profile 181
- RACF resource protection
 - deactivating or reactivating
 - using RVARY command 399
- RACF security topics
 - classroom courses xii
- RACF segment
 - group profile
 - changing 104
 - defining 24
 - displaying 217
 - suppressing display 217
 - user profile
 - changing 115
 - defining 49
 - displaying 227
 - suppressing display 229
- RACF TSO commands
 - compared to ISPF panels 16
 - entering
 - background 18
 - foreground 17
- RACFVARS class
 - description 509, 514
- RACGLIST class
 - description 509
- RACLINK command
 - description 296
 - examples 301
 - RACF requirements 296
 - syntax 297
- RACLIST operand
 - SETROPTS command 464
- RALTER command
 - description 303
 - examples 334
 - RACF requirements 304
 - syntax 305
- RDEFINE command
 - description 337
 - examples 368
- RDEFINE command (*continued*)
 - RACF requirements 338
 - syntax 340
- RDELETE command
 - description 371
 - examples 374
 - RACF requirements 372
 - syntax 372
- reactivating
 - RACF resource protection
 - using RVARY command 399
- REALDSN operand
 - SETROPTS command 467
- REALM class
 - description 512
- recording
 - RACROUTE REQUEST=VERIFY statistics 453
 - real data set names 467
 - statistics for resources 470
- REFRESH operand
 - SETROPTS command 467
- refreshing
 - generic profile checking 449
 - global access checking 451
 - in-storage generic profiles and program control
 - tables 467
- region size for TSO
 - maximum
 - for existing user 162
 - for new user 80
 - minimum
 - defining for user 81
 - for existing user 164
- REMOVE command
 - description 376
 - examples 378
 - RACF requirements 376
 - syntax 377
- REMOVE operand
 - RACDCERT command 280
- removing
 - authority to access a profile 249
 - names from access list 249
 - RACF protection from data set profile 181
 - user's access to system 158, 178
- RESET operand
 - PERMIT command 253
- RESET(ALL) operand
 - PERMIT command 253
- RESET(STANDARD) operand
 - PERMIT command 253
- RESET(WHEN) operand
 - PERMIT command 253
- RESGROUP operand
 - RLIST command 391
- resource access authority
 - specifying in access list 251
- resource group
 - adding
 - resource names as members 309

- resource profile
 - naming 495
- RESOWNER suboperand
 - ADDSD command 39
 - ALTDSD command 96
- RESTART command
 - description 379
 - example 381
 - RACF requirements 379
 - syntax 379
- restoring user's access to system 177
- RESTRICTED operand
 - ADDUSER command 78
 - ALTUSER command 157
- restrictions
 - logon for existing user 78, 157
- RESUME operand
 - ALTUSER command 158
 - CONNECT command 177
- RETAIN suboperand
 - DLFDATA operand 352
- RETPD operand
 - ADDSD command 43
 - ALTDSD command 99
 - SETROPTS command 467
- return codes 11
- REVOKE operand
 - ALTUSER command 158
 - CONNECT command 178
- REVOKE suboperand
 - PASSWORD operand 461
- revoking
 - user ID based on consecutive invalid passwords 461
 - user's access to system 158, 178
 - user's TSO authority 165
- RING operand
 - RACDCERT command 280, 281
- RLIST command
 - description 382
 - examples 393
 - RACF requirements 385
 - syntax 386
- RMTOPS class
 - description 512
- RODMMGR class
 - description 512
- ROLE class
 - description 512
- ROUTCODE suboperand
 - ADDUSER command 73
 - ALTUSER command 151
- RRSFDATA class
 - description 509
- RULEn suboperand
 - PASSWORD operand 461
- rules
 - establishing password syntax
 - ISPF panel 16
- RVARSMBR class
 - description 509, 514

- RVARY command
 - description 399
 - examples 405
 - RACF requirements 401
 - syntax 401
- RVARYPW operand
 - SETROPTS command 468

S

- sample
 - ISPF panel 17
- SAUDIT operand
 - SETROPTS command 468
- SCDMBR class
 - description 509, 514
- SCICSTST class
 - description 510
- SDNFILTER operand
 - RACDCERT command 283
- SDSF (System Display and Search Facility)
 - general resource class 509
- SDSF class
 - description 509
- SEARCH command
 - description 408
 - examples 418
 - RACF requirements 409
 - syntax 410
- searching for profiles in RACF database 408
- SECDATA class
 - description 509, 514
- SECLABEL class
 - description 509, 514
- SECLABEL operand
 - ADDSD command 43
 - ADDUSER command 78
 - ALTDSD command 99
 - ALTUSER command 159
 - DISPLAY command 195
 - RALTER command 323
 - RDEFINE command 359
 - SEARCH command 412
 - SIGNOFF command 478
- SECLABEL suboperand
 - ADDUSER command 81
- SECLABELAUDIT operand
 - SETROPTS command 468
- SECLEVEL operand
 - ADDSD command 44
 - ADDUSER command 78
 - ALTDSD command 99
 - ALTUSER command 160
 - RALTER command 323
 - RDEFINE command 359
 - SEARCH command 412
- SECLEVELAUDIT operand
 - SETROPTS command 469
- SECONDARY suboperand
 - ALTUSER command 134

- security category
 - adding to user profile 121
 - changing
 - for data set profile 93
 - general resource profile 308
 - defining 347
 - for data set profile 38
 - for general resource profile 343
 - for user profile 53, 65, 66, 67, 68, 69
 - deleting 311
 - from general resource profile 309
 - from user profile 122
 - undefined category names 309
 - searching on
 - for general resource profiles 412
- security classification of users and data
 - defining categories and levels 347
 - in data set profile 38, 94
 - in general resource profile 309, 344
 - in user profile 53, 65, 67, 68, 69, 122
- security label
 - changing
 - for user profile 159
 - defining
 - for user profile 78
 - for new user 81
 - for TSO 81
 - for user profile 43, 359
 - translating on inbound jobs or SYSOUT 348
- security level
 - changing
 - for data set profile 99
 - defining 347
 - deleting 311
 - for existing data set profile 99
 - for existing general resource profile 323
 - for existing user profile 160
 - for new data set profile 44
 - for new general resource profile 359
 - for new user profile 78
 - for resource profile 323
 - in user profile 160
 - logging access attempts based on 469
 - searching on
 - general resource profiles 412
 - using as search criteria 412
- security retention period
 - for existing tape data set profile 99
 - for new tape data set profile 43
 - system-wide for tape data sets 467
- security topics for RACF
 - classroom courses xii
- segment
 - CICS segment
 - displaying for a user profile 227
 - for new user profile 54
 - DFP segment
 - changing in group profile 107
 - changing in user profile 128
 - displaying for a user profile 228
 - displaying for group profile 217
- segment (*continued*)
 - DFP segment (*continued*)
 - existing user profile, deleting from 129
 - for new group profile 26
 - for new user profile 58
 - DLFDATA segment
 - authority to define 339
 - EIM segment
 - displaying for a user profile 228
 - KERB segment
 - displaying for a user profile 228
 - LANGUAGE segment
 - displaying for a user profile 228
 - LNOTES segment
 - changing in user profile 134
 - displaying for a user profile 229
 - NDS segment
 - changing in user profile 135
 - displaying for a user profile 229
 - NETVIEW segment
 - changing in user profile 136
 - displaying for group profile 229
 - OMVS segment
 - changing in group profile 109
 - changing in user profile 139
 - displaying for group profile 217
 - displaying for user profile 229
 - for new user profile 65
 - OPERPARM segment
 - displaying for user profile 230
 - for new user profile 70
 - OVM segment
 - changing in group profile 111
 - displaying for user profile 230
 - PROXY segment
 - displaying for a user profile 230
 - RACF segment
 - changing for a group profile 104
 - defining for group profile 24
 - displaying for group profile 217
 - displaying for user profile 227
 - for existing user profile 115
 - for new user profile 49
 - suppressing display for group profile 217
 - suppressing display for user profile 229
 - SESSION segment
 - displaying for general resource profile 392
 - for general resource profile 324
 - for general resource profiles 360
 - SSIGNON segment
 - displaying 392
 - STDATA segment
 - defining 362
 - modifying 326
 - TSO segment
 - deleting from user profile 165
 - displaying for a user profile 231
 - for existing user profile 160
 - for new user profile 79
 - WORKATTR segment
 - displaying for group profile 231

- SERVAUTH class
 - description 509
- SERVER class
 - description 509
- SESSION operand
 - RALTER command 324
 - RDEFINE command 360
 - RLIST command 392
- SESSION segment
 - displaying
 - general resource profile 392
- SET command
 - Callable Service Trace Types 429
 - description 420
 - examples 432
 - PDCallable Service Trace Types 431
 - RACF requirements 420
 - Racroute Trace Types 431
 - syntax 420
- SET operand
 - ADDSD command 42
 - ALTDSD command 97
 - DELDSD command 184
- SETONLY operand
 - ADDSD command 42
- SETOPTS command
 - description 435
 - examples 471
 - RACF requirements 437
 - syntax 438
- SFSCMD class
 - description 514
- Shared File System Profiles 504
- shared in-storage profile
 - SETOPTS GENLIST processing for 450
 - SETOPTS RACLIST processing for 464
- SHARED suboperand
 - ALTGROUP command
 - for GID suboperand 29, 110
 - UID suboperand
 - ADDUSER command 66
 - ALTUSER command 141
- shortcut keys 517
- signed-on-from list 193, 476
- SIGNOFF command
 - description 476
 - examples 478
 - RACF requirements 476
 - syntax 477
- SIGNON operand
 - DISPLAY command 194
- SIGNWITH operand
 - RACDCERT command 274
- SIMS class
 - description 511
- SINGLEDSN operand
 - RALTER command 325
 - RDEFINE command 361
- SITE operand
 - RACDCERT command 261
- SIZE operand
 - RACDCERT command 273
- SIZE suboperand
 - ADDUSER command 81
 - ALTUSER command 164
- SMESSAGE class
 - description 509
- SMS
 - general resource classes 512
- SNAME suboperand
 - ALTUSER command 134
- SOMDOBJs class
 - description 509
- SPECIAL attribute
 - for existing user 160
 - for new user 79
 - logging activities for 468
- SPECIAL operand
 - ADDUSER command 79
 - ALTUSER command 160
 - CONNECT command 179
- Specifying the distinguished name the LDAP proxy server will use
 - ADDUSER command 77
 - ALTUSER command 156
 - RALTER command 322
 - RDEFINE command 358
- Specifying the password the LDAP proxy server will use
 - ADDUSER command 77
 - ALTUSER command 157
 - RALTER command 322
 - RDEFINE command 358
- Specifying the URL of the LDAP proxy server
 - ADDUSER command 76
 - ALTUSER command 155
 - RALTER command 321
 - RDEFINE command 357
- SRDIR 1
- SRFILE 1
- SSIGNON operand
 - KEYENCRYPTED suboperand 326, 362
 - KEYMASKED suboperand 326, 361
 - RALTER command 325, 326
 - RDEFINE command 361, 362
 - RLIST command 392
- SSIGNON segment
 - displaying
 - general resource profile 392
- standard access list 247
- STARTED class
 - description 509
- started task
 - security category checking
 - RALTER command 309
 - RDEFINE command 344
 - security level checking 324
- statistics
 - bypassing recording of 453
 - displaying
 - for data set profile 208
 - general resource profile 392

- statistics (*continued*)
 - recording
 - for classes 470
 - for REQUEST=VERIFY processing 453
- STATISTICS operand
 - LISTDSD command 208
 - RLIST command 392
 - SETROPTS command 470
- STATUS suboperand
 - RVARYPW operand (SETROPTS command) 468
- STDATA operand
 - RALTER command 326
 - RDEFINE command 362
 - RLIST command 392
- STDATA segment
 - displaying
 - general resource profile 392
 - RALTER command 326
 - RDEFINE command 362
- STOP command
 - description 479
 - examples 480
 - RACF requirements 479
 - syntax 479
- storage class for DFP
 - adding
 - to new group profile 27
 - changing
 - for group profile 108
 - in user profile 129
 - defining
 - for user profile 58
- STORAGE suboperand
 - ADDUSER command 73
 - ALTUSER command 151
- STORCLAS class
 - description 512
- STORCLAS sub-operand
 - TARGET command 490
- STORCLAS suboperand
 - ADDGROUP command 27
 - ADDUSER command 58
 - ALTGROUP command 108
 - ALTUSER command 129
- SUBJECTSDN operand
 - RACDCERT command 272
- SUBSYSNM class
 - description 512
- subsystem prefix
 - ADDGROUP command 26
 - ADDSD command 36
 - ADDUSER command 52
 - ALTDSD command 92
 - ALTGROUP command 106
 - ALTUSER command 121
 - CONNECT command 174
 - DELDSD command 183
 - DELGROUP command 187
 - DELUSER command 191
 - DISPLAY command 194
 - LISTDSD command 204
- subsystem prefix (*continued*)
 - LISTGRP command 216
 - LISTUSER command 226
 - PASSWORD command 244
 - PERMIT command 250
 - RACLINK command 298
 - RALTER command 307
 - RDEFINE command 341
 - RDELETE command 373
 - REMOVE command 377
 - RESTART command 380
 - RLIST command 387
 - RVARY command 401
 - SEARCH command 410
 - SET command 421
 - SETROPTS command 440
 - SIGNOFF command 477
 - STOP command 480
 - TARGET command 482
- summary
 - of RACF authorities 1
 - of RACF commands 1
- superior group
 - for existing group 112
 - for new group 30
- SUPGROUP operand
 - ADDGROUP command 30
 - ALTGROUP command 112
- suppressing display of RACF segment
 - group profile 217
 - user profile 229
- SURROGAT class
 - description 509
- SVFMR operand
 - RALTER command 328
 - RDEFINE command 363
 - RLIST command 392
- SWITCH operand
 - RVARY command 403
- SWITCH suboperand
 - RVARYPW operand (SETROPTS command) 468
- SYNTAX operand
 - HELP command 199
- syntax rules
 - for commands 9
 - for passwords 461
- SYSMVIEW class
 - description 509
- SYSNAME operand
 - TARGET command 487
- SYSOUT class for TSO
 - for existing user 164
 - for new user 82
- SYSOUT data set destination
 - for existing user 162
 - for new user 80
- SYSOUTCLASS suboperand
 - ADDUSER command 82
 - ALTUSER command 164
- sysplex communication
 - data sharing option 4

- sysplex communication (*continued*)
 - example of RACGLIST processing 368
 - example of RVAR Y NODATASHARE 405
 - logging RVAR Y commands 399
 - RVAR Y ACTIVE command 402
 - RVAR Y INACTIVE command 402
 - RVAR Y SWITCH command 404
- System Display and Search Facility (SDSF)
 - See SDSF (System Display and Search Facility)
- system message reception
 - for new user profile 73
 - for user profile 150
- system-wide options
 - activating 435
 - displaying current RACF 454, 456
 - example of display 475

T

- tape data set
 - creating entry in TVTOC 42
 - defining a new profile to protect 41
 - file sequence number 40
 - searching for profile 411
 - security retention period for existing profile 99
 - security retention period for new profile 43
 - specifying tape volume to contain single data set 325, 361
 - system-wide security retention period 467
- tape data set protection
 - activating or deactivating 470
- TAPE operand
 - ADDSD command 41
 - SEARCH command 411
- tape volume
 - creating TVTOC for 331, 365
 - deactivating protection 403
 - displaying volume information for 417
 - searching for expired 412
 - specifying to contain single data set 325, 361
- TAPEDSN operand
 - SETROPTS command 470
- TAPEVOL class
 - description 509, 514
- TAPEVOL profile
 - changing to nonautomatic 248
- TARGET command
 - description 481
 - examples 491
 - RACF requirements 481
 - syntax 481
- TCICSTRN class
 - description 510
- TEMPDSN class
 - description 509
- terminal
 - limiting access to 333, 366
 - time zone 329, 364
 - UACC for undefined terminals 470
- terminal authorization checking
 - for users in a new group 30

- terminal authorization checking (*continued*)
 - for users in an existing group 113
- TERMINAL class
 - description 509, 514
- terminal id
 - syntax 11
- TERMINAL operand
 - SETROPTS command 470
- TERMUACC operand
 - ADDGROUP command 30
 - ALTGROUP command 113
- THREADSMAX operand
 - ADDUSER command 69
- THREADSMAX suboperand
 - ALTUSER command 144
- time of day
 - existing user can access system 167
 - new user can access system 83
 - terminal can access system 333, 367
- TIME operand
 - ADDUSER command 82
 - ALTUSER command 166
 - RACDCERT command 274
 - RALTER command 333
 - RDEFINE command 366
- TIMEOUT suboperand
 - ADDUSER command 54
 - ALTUSER command 124
- TIMEZONE operand
 - RALTER command 329
 - RDEFINE command 364
- TIMS class
 - description 511
- Tivoli
 - general resource class 512
- Tivoli Service Desk
 - general resource classes 511
- TME operand
 - ADDGROUP command 30
 - ADDSD command 44
 - ALTDSD command 100
 - ALTGROUP command 113
 - LISTDSD command 208
 - LISTGRP command 392
 - RALTER command 329
 - RDEFINE command 364
 - RLIST command 218
- TMEADMIN class
 - description 512
- TRACE operand
 - RALTER command 327
 - RDEFINE command 363
 - SET command 427
- TRUST operand
 - RACDCERT command 266, 270, 286
- TRUSTED operand
 - RALTER command 328
 - RDEFINE command 363
- TSO default prefix for data set 17

- TSO logon information
 - changing
 - default for user profile 160
 - defining
 - default for user profile 79
 - deleting
 - from user profile 165
- TSO operand
 - ADDUSER command 79
 - ALTUSER command 160
 - LISTUSER command 231
- TSO segment
 - deleting from user profile 165
 - displaying for user profile 231
 - for existing user profile 160
 - for new user profile 79
- TSO SUBMIT command 18
- TSO/E
 - general resource classes 513
- TSOAUTH class
 - description 513
- TSOPROC class
 - description 513
- TVTOC (tape volume table of contents)
 - creating entry for tape data set 42
- TVTOC operand
 - RALTER command 331
 - RDEFINE command 365
 - RLIST command 392
- type of broadcast messages
 - for new user profile 71
 - for user profile 148

U

- UACC (universal access authority)
 - changing
 - default for user profile 165
 - default in user's connect profile 179
 - for data set profile 101
 - for general resource profile 332
 - defining
 - default for user profile 82
 - default in user's connect profile 179
 - for data set profile 45
 - for general resource profile 365
 - for undefined terminals 470
- UACC operand
 - ADDSD command 45
 - ADDUSER command 82
 - ALTDSD command 101
 - ALTUSER command 165
 - CONNECT command 179
 - RALTER command 332
 - RDEFINE command 365
- UAUDIT operand
 - ALTUSER command 166
- UCICSTST class
 - description 510
- UD suboperand
 - ADDUSER command 74
- UD suboperand (*continued*)
 - ALTUSER command 152
- UID operand
 - SEARCH command 416
- UID suboperand
 - ADDUSER command 65
 - ALTUSER command 140
- UIMS class
 - description 511
- UNAME suboperand
 - ALTUSER command 135
- undelivered message reception
 - changing for user profile 152
 - defining for user profile 74
- unit devices for TSO
 - for existing user 165
 - for new user 82
- UNIT operand
 - ADDSD command 45
 - ALTDSD command 101
- UNIT suboperand
 - ADDUSER command 82
 - ALTUSER command 165
- unit type
 - changing for data set profile 101
 - defining for data set profile 45
- universal group
 - for new group 30
- UNIVERSAL operand
 - ADDGROUP command 30
- UNIXMAP class
 - description 513
- UNIXPRIV class
 - description 513
- URI operand
 - RACDCERT command 277
- USAGE operand
 - RACDCERT command 280
- USE group authority
 - description 13
- user
 - limiting access to system 82, 166
- user data for TSO
 - changing 165
 - defining 82
- user ID
 - as new owner
 - of data set profiles of removed user 378
 - as owner
 - of connect profile 177
 - of data set profile 98
 - of general resource profile 321
 - of group profile 112
 - of new data set profile 42
 - of new general resource profile 357
 - of new group profile 30
 - of new user profile 75
 - of user profile 154
 - changing access to resource for 252
 - deactivating an unused 453
 - displaying data set profiles for 205

- user ID (*continued*)
 - displaying user profile for 227
 - no password 76, 155
 - protected 76, 155
 - removing user from group 377
 - revoking based on consecutive invalid passwords 461
 - syntax 10
 - to add new user profile 52
 - to alter user profile 121
 - to change password 245
 - to connect to group 175
 - to receive notify message
 - for existing data set profile 98
 - for general resource profile 320, 357
 - for new data set profile 42
 - translating on inbound jobs or SYSOUT 348
 - when deleting user profile 191
- user name
 - changing 135
 - defining 62
- USER operand
 - DISPLAY command 195
 - PASSWORD command 245
 - RALTER command
 - =MEMBER operand 326
 - RDEFINE command
 - =MEMBER operand 362
 - SEARCH command 416
 - SIGNOFF command 477
- user profile
 - changing 115
 - defining 48
 - deleting 190
 - displaying 223
 - RACF segment 49
 - removing from group 376
 - searching for based on last reference 411
- USERDATA suboperand
 - ADDUSER command 82
 - ALTUSER command 165
- userid-named data set
 - activating model profile for 459

V

- valid values
 - FORMAT operand 277
- VCICSCMD class
 - description 510
- VMBATCH class
 - description 514
- VMBR class
 - description 514
- VMCMD class
 - description 514
- VMEVENT class
 - description 514
- VMMAC class
 - description 514

- VMMDISK class
 - description 514
- VMNODE class
 - description 514
- VMPOSIX class
 - description 515
- VMRDR class
 - description 514
- VMSEGMT class
 - description 514
- VMXEVENT class
 - description 515
- VOLUME operand
 - ADDSD command 45
 - ALTDSD command 101
 - DELDSD command 184
 - LISTDSD command 208
 - PERMIT command 254
 - SEARCH command 417
- volume serial
 - syntax 11
- volume serial number
 - adding volume to tape volume profile 311
 - deleting a data set 184
 - deleting volume from tape volume profile 311
 - displaying data set profile 208
 - for controlled program 349
 - for existing data set 101
 - for multivolume data set 94
 - for new data set 45
 - specifying when creating access list 254
 - using as search criteria 417
 - using to locate model profile 40, 354
- VOLUME sub-operand
 - TARGET command 490
- VSAM data set
 - protecting 500
 - searching for 411
- VSAM operand
 - SEARCH command 411
- VSAMDSET as a high-level qualifier 34
- VTAM (Virtual Telecommunications Access Method)
 - general resource class 509
- VTAMAPPL class
 - description 509
- VXMBR class
 - description 514

W

- WAACCNT suboperand
 - ADDUSER command 83
 - ALTUSER command 167
- WAADDRx suboperand
 - ADDUSER command 83
- WABLDG suboperand
 - ADDUSER command 84
 - ALTUSER command 168
- WADEPT suboperand
 - ADDUSER command 84
 - ALTUSER command 168

- WANAME suboperand
 - ADDUSER command 84
 - ALTUSER command 168
- warning indicator
 - searching for resources with 412
- warning message
 - number of days before password expires 462
- WARNING operand
 - ADDSD command 45
 - ALTDSD command 101
 - RALTER command 332
 - RDEFINE command 366
 - SEARCH command 412
- WARNING suboperand
 - PASSWORD operand 462
- WAROOM suboperand
 - ADDUSER command 84
 - ALTUSER command 169
- WDSQUAL operand
 - TARGET command 488
- WHEN DAYS operand
 - ADDUSER command 82
 - ALTUSER command 166
 - RALTER command 333
 - RDEFINE command 366
- WHEN TIME operand
 - ADDUSER command 82
 - ALTUSER command 166
 - RALTER command 333
 - RDEFINE command 366
- WHEN(APPSPORT) operand
 - PERMIT command 254
- WHEN(CONSOLE) operand
 - PERMIT command 254
- WHEN(JESINPUT) operand
 - PERMIT command 254
- WHEN(PROGRAM) operand
 - PERMIT command 254
 - SETROPTS command 471
- WHEN(SYSID) operand
 - PERMIT command 255
- WHEN(TERMMINAL) operand
 - PERMIT command 255
- WIMS class
 - description 511
- WITHLABEL operand
 - RACDCERT command 268, 274, 285
- WORKATTR operand
 - ADDUSER command 83
 - ALTUSER command 167, 169
 - LISTUSER command 231
- WORKATTR segment
 - group profile
 - displaying 231
- WORKSPACE operand
 - TARGET command 489
- WRITER class
 - description 509, 515

Z

- z/OS UNIX
 - general resource classes 513

Readers' Comments — We'd Like to Hear from You

z/OS
Security Server RACF
Command Language Reference

Publication No. SA22-7687-03

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.



Cut or Fold
Along Line

Fold and Tape

Please do not staple

Fold and Tape



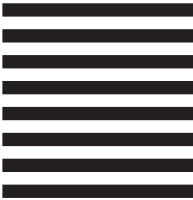
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY
12601-5400



Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold
Along Line



Program Number: 5694-A01, 5655-G52

Printed in U.S.A.

SA22-7687-03

